Filesystem Hierarchy Standard
por Filesystem Hierarchy Standard Group
Editado por Rusty Russell, Daniel Quinlan, e Christopher Yeoh

Publicado em 28 de janeiro de 2004


Este padrão consiste em um conjunto de requisitos e orientações para a colocação de arquivos e diretórios em sistemas operacionais Unix-like. As diretrizes são destinadas a apoiar a interoperabilidade de aplicações, ferramentas de administração do sistema, ferramentas de desenvolvimento, e roteiros, bem como uma maior uniformidade da documentação para esses sistemas.

# Índice


Introdução........ 1
1.1. Finalidade........ 1
1.2. Convenções........ 1
2. O Filesystem........ 2
3. O sistema de arquivos raiz........ 3
3.1. Finalidade........ 3
3.2. Requisitos........ 4
3.3. Opções Específicas........ 4
3.4. / bin: binários usuário Essencial de comando (para utilização por todos os usuários)........ 4
3.4.1. Finalidade........ 4
3.4.2. Requisitos........ 5
3.4.3. Opções Específicas........ 6
3.5. / boot: arquivos estáticos do carregador de inicialização........ 6
3.5.1. Finalidade........ 6
3.5.2. Opções Específicas........ 7
3.6. / dev: Arquivos de dispositivos........ 7
3.6.1. Finalidade........ 7
3.6.2. Opções Específicas........ 7
3.7. / etc: Configuração do sistema específico........ 7
3.7.1. Finalidade........ 7
3.7.2. Requisitos........ 7
3.7.3. Opções Específicas........ 7
3.7.4. / etc / opt: Arquivos de configuração para / opt........ 8
3.7.4.1. Finalidade........ 9
3.7.4.2. Requisitos........ 9
3.7.5. / etc/X11: Configuração para o X Window System (opcional)........ 9
3.7.5.1. Finalidade........ 9
3.7.5.2. Opções Específicas........ 9
3.7.6. / etc / sgml /: arquivos de configuração para SGML (opcional) ........ 9
3.7.6.1. Finalidade........ 10
3.7.7. / etc / xml: Os arquivos de configuração para XML (opcional)........ 10
3.7.7.1. Finalidade........ 10
3.8. / home: diretório home do usuário (opcional)........ 10
3.8.1. Finalidade........ 10
3.8.2. Requisitos........ 10
3.9. / lib Bibliotecas essenciais partilhadas e módulos do kernel........ 10
3.9.1. Finalidade........ 10
3.9.2. Requisitos........ 10
3.9.3. Opções Específicas........ 11
3.10. bibliotecas / lib <qual>: formato alternativo essencial compartilhada (opcional)........ 11
3.10.1. Finalidade........ 11
3.10.2. Requisitos........ 11
3.11. / media: Ponto de montagem para mídias removíveis........ 11

3.11.1. Finalidade........ 11
3.11.2. Opções Específicas........ 12
3.12. / mnt: Ponto de montagem para um sistema de arquivos montados temporariamente........ 12
3.12.1. Finalidade........ 12
3.13. / opt: Add-on pacotes de aplicativos de software........ 12
3.13.1. Finalidade........ 12





3.13.2. Requisitos........ 12
3.14. / root: Diretório home do usuário root (opcional)........ 13
3.14.1. Finalidade........ 14
3.15. / sbin: Sistema binários........ 14
3.15.1. Finalidade........ 14
3.15.2. Requisitos........ 14
3.15.3. Opções Específicas........ 14
3.16. / srv: Dados para serviços fornecidos por este sistema........ 15
3.16.1. Finalidade........ 15
3.17. / tmp: Arquivos temporários........ 15
3.17.1. Finalidade........ 15
4. A hierarquia / usr........ 18
4.1. Finalidade........ 18
4.2. Requisitos ........ 18
4.3. Opções Específicas........ 18
4.4. / usr/X11R6: X Window System, Version 11 Release 6 (opcional)........ 18
4.4.1. Finalidade........ 19
4.4.2. Opções Específicas........ 19
4.5. / usr / bin: A maioria dos comandos do usuário........ 19
4.5.1. Finalidade........ 19
4.5.2. Opções Específicas........ 19
4.6. / usr / include: Diretório padrão para arquivos de inclusão........ 20
4.6.1. Finalidade........ 20
4.6.2. Opções Específicas........ 20
4.7. / usr / lib: Bibliotecas para programação e pacotes........ 20
4.7.1. Finalidade........ 20
4.7.2. Opções Específicas........ 20
4.8. / usr / lib <qual>: bibliotecas formato alternativo (opcional)........ 21
4.8.1. Finalidade........ 21
4.8.2. / usr /local: Hierarquia local........ 21
4.8.2.1. Finalidade........ 21
4.8.2.2. Requisitos........ 21
4.8.2.3. Opções Específicas........ 21
4.9. / usr / local / share........ 22
4.10. / usr / sbin: binários não-essenciais do sistema padrão........ 22
4.10.1. Finalidade........ 22
4.11. / usr / share: dados independentes de arquitetura........ 22
4.11.1. Finalidade........ 22
4.11.2. Requisitos........ 22
4.11.3. Opções Específicas........ 23
4.11.4. / usr / share / dict: Listas de palavras (opcional)........ 23
4.11.4.1. Finalidade........ 23
4.11.4.2. Opções Específicas........ 23
4.11.5. / usr / share / man: Paginas de manual........ 24

4.11.5.1. Finalidade........................................................................................... 24
4.11.5.2. Opções Específicas........................................................................... 25
4.11.6. / usr / share / misc: Dados de diversas arquiteturas independentes.................................. 26
4.11.6.1. Opções Específicas........................................................................... 26
4.11.7. / usr / share / sgml: dados SGML (opcional)........................................................ 27
4.11.7.1. Finalidade........................................................................................... 27
4.11.7.2. Opções Específicas........................................................................... 27



4.11.8. / usr / share / xml: dados XML (opcional)........................................................ 27
4.11.8.1. Finalidade........................................................................................... 27
4.11.8.2. Opções Específicas........................................................................... 27
4.12. / usr / src: código-fonte (opcional)........................................................................ 28
4.12.1. Finalidade............................................................................................... 28
5. A hierarquia / var................................................................................................... 30
5.1. Finalidade.............................................................................................................. 30
5.2. Requisitos.............................................................................................................. 30
5.3. Opções Específicas................................................................................................ 30
5.4. / var / account: processo de contabilização de logs (opcional)....................................... 31
5.4.1. Finalidade............................................................................................... 31
5.5. / var / cache: Dados de aplicativos de cache........................................................... 31
5.5.1. Finalidade............................................................................................... 31
5.5.2. Opções Específicas................................................................................... 31
5.5.3. / var / cache / fonts: fontes gerados localmente (opcional)........................................ 32
5.5.3.1. Finalidade........................................................................................... 32
5.5.3.2. Opções Específicas........................................................................... 32
5.5.4. / var / cache / man: Páginas de manuais formatados localmente (opcional)......................... 32
5.5.4.1. Finalidade........................................................................................... 32
5.6. / var / crash: despejos de memória do sistema (opcional)............................................ 33
5.6.1. Finalidade............................................................................................... 33
5.7. / var / games: Dados de jogos variados (opcional).................................................... 33
5.7.1. Finalidade............................................................................................... 33
5.8. / var / lib: informações sobre o estado das variáveis.................................................. 33
5.8.1. Finalidade............................................................................................... 33
5.8.2. Requisitos............................................................................................... 34
5.8.3. Opções Específicas................................................................................... 34
5.8.4. / var / lib / <editor> Editor de arquivos de backup e de estado (opcional)........................... 34
5.8.4.1. Finalidade........................................................................................... 34
5.8.5. / var / lib / hwclock: Estado diretório para hwclock (opcional)...................................... 35
5.8.5.1. Finalidade........................................................................................... 35
5.8.6. / var / lib / misc: Dados variáveis diversos............................................................ 35
5.8.6.1. Finalidade........................................................................................... 35
5.9. / var / lock: Arquivos bloqueados........................................................................... 35
5.9.1. Finalidade............................................................................................... 35
5.10. / var / log: Log de arquivos e diretórios.................................................................. 35
5.10.1. Finalidade............................................................................................. 36
5.10.2. Opções Específicas ................................................................................. 36
5.11. / var / mail: Arquivos de caixa de correio (opcional)................................................. 36
5.11.1. Finalidade............................................................................................. 36
5.12. / var / opt: dados variáveis para / opt ................................................................... 36
5.12.1. Finalidade............................................................................................. 36

5.13. / var / run: Dados variáveis de Run-Time 37
5.13.1. Finalidade 37
5.13.2. Requisitos 37
5.14. / var / spool: Aplicação de dados de impressão 37
5.14.1. Finalidade 37
5.14.2. Opções Específicas 37
5.14.3. / var / spool / lpd: Impressoras de linha e filas de impressão (opcional) 38
5.14.3.1. Finalidade 38
5.14.3.2. Opções Específicas 38



5.14.4. / var / spool / rwho: arquivos Rwhod (opcional) 38
5.14.4.1. Finalidade 38
5.15. / var / tmp: Arquivos temporários preservados entre reinicializações do sistema 38
5.15.1. Finalidade 38
5.16. / var / yp: Network Information Service (NIS) arquivos de dados (opcional) 39
5.16.1. Finalidade 39
6. Sistema Operacional anexo específico 40
6.1. Linux 40
6.1.1. /: Diretório raiz 40
6.1.2. / bin: Comandos binários essenciais de usuário (para utilização por todos os usuários) 40
6.1.3. / dev: Dispositivos e arquivo especiais 40
6.1.4. / etc: Configurações específicas do sistema 40
6.1.5. / lib64 e / lib32: bibliotecas 64/32-bit (arquitetura dependente) 41
6.1.6. / proc: Kernel e informações de processo do sistema de arquivos virtual 41
6.1.7. / sbin: Sistemas binários essenciais 41
6.1.8. / usr / include: arquivos de cabeçalho incluídos por programas C 42
6.1.9. / usr / src: códigos fonted 42
6.1.10. / var / spool / cron: cron e postos de trabalho 43
7. Apêndice 44
7.1. A lista de discussão FHS 44
7.2. Antecedentes da FHS 44
7.3. Orientações Gerais 44
7.4. Âmbito 44
7.5. Agradecimentos 45
7.6. Colaboradores 45

# Capítulo 1. Introdução.

## 1.1. - Propósito

### Esta norma permite:

- Software para prever a localização de arquivos e diretórios instalados, e
- Usuários para prever a localização de arquivos e diretórios instalados.

### Nós Fazemos isso para:

- Especificar os princípios orientadores para cada área do sistema de arquivos,
- Especificar os arquivos e diretórios mínimo exigido,
- Enumerar exceções aos princípios e
- Enumerar os casos específicos em que houve conflito histórico.

### O documento FHS é usado por:

Fornecedores independentes de software para criar aplicações que são compatíveis com FHS, e trabalhar com as distribuições que são compatíveis com FHS.

- Criadores do sistema operacional para fornecer sistemas que são FHS compatível, e
- Usuários para entender e manter o compatibilidade FHS de um sistema.

### O documento FHS tem um alcance limitado:

- Local de colocação de arquivos locais é uma questão local, de modo FHS não tenta usurpar os administradores do sistema.
- FHS aborda questões onde colocações arquivo deve ser coordenada entre os vários partidos, como local sites, distribuições, aplicações, documentação, etc.

## 1.2. - Convenções

Recomendamos que você leia uma versão typeset deste documento e não a versão em texto puro. Na versão typeset , os nomes dos arquivos e diretórios são exibidos em uma fonte de largura constante.

Componentes de nomes de arquivos que variam são representados por uma descrição do conteúdo incluído em caracteres "<" e ">" caracteres, <thus>. Endereços de correio eletrônico também são colocados entre "<" e ">", mas são mostrados na forma habitual.

Componentes opcionais de nomes de arquivos são colocados entre caracteres "[" e "]" e pode ser combinado com a convenção "<" e ">". Por exemplo, se um nome de arquivo pode ocorrer com ou sem uma extensão, pode ser representada por <filename> [. <extension>].

Variáveis Substrings de nomes de diretório e nomes de arquivos são indicados por "*".

As seções do texto marcado como Fundamentação são explicativas e não são normativas.



# Capítulo 2. - O Filesystem (sistema de arquivos)

Esta norma pressupõe que o sistema operacional subjacente um sistema de arquivos FHS compatível suportaa as mesmas características básicas de segurança encontrada na maioria dos sistemas de arquivos UNIX.

É possível definir duas distinções independentes entre arquivos: compartilhável vs não-compartilhável e variável vs estática.

Em geral, os arquivos que diferem entre si em qualquer um destes aspectos, devem estar localizados em diretórios diferentes. Isto torna mais fácil para armazenar arquivos com diferentes características de uso em sistemas de arquivos diferentes.

- Arquivo "Compartilhável" são aqueles que podem ser armazenados em um host e utilizada em outro.
- Arquivo "Não-Compartilhável" são aqueles que não são utilizáveis de forma mútua. Por exemplo, os arquivos no diretório home do usuário são compartilháveis enquanto arquivos de bloqueio do dispositivo não são.

- Arquivos "Static" inclui arquivos binários, bibliotecas, arquivos de documentação e outros arquivos que não mudam sem a intervenção do administrador de sistema.
- Arquivo "Variável" são arquivos que não são estáticos.

### Fundamentação.

Arquivos Compartilháveis podem ser armazenados em um host e usado em vários outros. Normalmente, no entanto, nem todos os arquivos no sistema de hierarquia de arquivos são compartilháveis e assim cada sistema de armazenagem local pode conter arquivos não compartilháveis .

É conveniente se todos os arquivos de um sistema exigirem que sejam armazenados em um host externo pode ser disponibilizado por uma montagem única ou por alguns diretórios do anfitrião estrangeiro.

Arquivos estáticos e variáveis devem ser separadas, porque arquivos estáticos, ao contrário de arquivos variável, pode ser armazenado em mídia somente de leitura e não precisam ser apoiados na mesma agenda que os arquivos variável.

Historicamente sistemas de arquivos hierárquicos UNIX-like contiveram ambos, estático e variável em ambos os locais / usr e / etc.
A fim de perceber as vantagens acima mencionadas, a hierarquia / var foi criada e todos os arquivos variáveis foram transferidos de / usr para / var. Consequentemente / usr agora pode ser montado read-only (se for um sistema de arquivos separado). Arquivos variáveis foram transferidos de / etc para / var durante um longo período como a tecnologia tem permitido.

Aqui está um exemplo de um sistema FHS compatível. (Outros esquemas FHS compatíveis são possíveis.)

| | Compartilhável | Não-compartilhável |
|---|---|---|
| **Estático** | **/usr**<br>**/opt** | **/etc**<br>**/boot** |
| **Variável** | **/var/mail**<br>**/var/spool/news** | **/var/run**<br>**/var/lock** |



# Capítulo 3. - O sistema de arquivos root

## 3.1 – Propósito.

O conteúdo do sistema de arquivos raiz deve ser adequado para iniciar, restaurar, recuperar e / ou reparar o sistema.

- Para iniciar um sistema, o suficiente deve estar presente na partição de root para montar outros sistemas de arquivos. Isto inclui: utilitários, configuração, informações iniciais de boot loader, e outros dados essenciais para iniciar o processo. / usr, / opt, e / var são concebidos de modo que eles podem estar localizados em outras partições e sistemas de arquivos.

- Para permitir a recuperação e / ou reparação de um sistema, os utilitários necessários para um mantenedor experiente para diagnosticar e reconstruir um sistema danificado deve estar presente no sistema de arquivos raiz.

- Para restaurar um sistema, as instalações necessárias para restaurar a partir de backups do sistema (em disquetes, fitas, etc) devem ser presentes no sistema de arquivos raiz.

## Fundamentação.

A preocupação primária utilizada para equilibrar estas considerações, que favorecem a colocar muitas coisas no sistema de arquivos raiz, é o objetivo de manter a raiz tão pequeno quanto possível. Por várias razões, é desejável manter o sistema de arquivos raiz de pequeno porte:

- É ocasionalmente montado a partir de mídia muito pequeno.

- O sistema de arquivos raiz contém muitos arquivos de configuração específicos. Possíveis exemplos incluem um kernel que é específico para o sistema, uma máquina específica, etc Isso significa que o sistema de arquivos raiz não é sempre compartilhável entre os sistemas em rede. Mantendo-o pequeno em servidores em sistemas em rede minimiza o quantidade de espaço perdido para as áreas de arquivos não compartilháveis. Ele também permite que estações de trabalho tenham unidades de disco rígido local menores.

- Embora você possa ter o sistema de arquivos raiz em uma partição grande, e pode ser capaz de preenchê-lo para o seu conteúdo, haverá pessoas com partições pequenas. Se você tem mais arquivos instalados, você pode encontrar incompatibilidades com outros sistemas usando sistemas de arquivos raiz em partições menores. Se você é um desenvolvedor então você pode estar transformando sua suposição em um problema para um grande número de usuários.

- Erros de disco que corrompem dados no sistema de arquivos raiz é um problema maior do que erros em qualquer outro partição. Um sistema de arquivos raiz pequeno é menos propenso à corrupção como o resultado de uma falha do sistema.

Aplicações nunca devem criar ou exigir arquivos especiais ou subdiretórios no diretório raiz. Outros locais naa hierarquia FHS fornecem flexibilidade mais do que suficiente para qualquer pacote.

## Fundamentação.

Há várias razões para ser proibido cria um novo subdiretório no sistema de arquivos raiz:

- Exige espaço em uma partição raiz que o administrador do sistema pode querer manter pequeno e simples para desempenhos ou razões de segurança.

- Ele evita o que determina o administrador do sistema, que pode ter configurado a distribuição de arquivos entre hierarquias padrões entre volumes montáveis.

Distribuições não deve criar novos diretórios na hierarquia de raiz, sem considerar com extremo cuidadoso as conseqüências, inclusive para a portabilidade dos aplicativos.





## 3.2 – Requisitos.

Os seguintes diretórios, ou links simbólicos para diretórios, são requeridos no /

| Diretório | Descrição |
|---|---|
| bin | Comandos essenciais binários |
| boot | Arquivos estáticos ou de inicialização |
| dev | Arquivos de dispositivos |
| etc | Sistema de configuração do Host especifico |
| lib | Bibliotecas essenciais compartilhadas e módulos do kernel |
| media | Ponto de montagem para mídia removível |
| mnt | Ponto de montagem para montar um sistema de arquivos temporáriamente |
| opt | Adiçao de pacotes de software de aplicação |
| sbin | Sistema binário essencial |
| srv | Dados para serviços providos por esse sistema |
| tmp | Arquivos temporários |
| usr | Hierarquia secundária |
| var | Dados variáveis |

Cada diretório listado acima é especificado em pormenor nas subseções abaixo. / usr e / var têm cada um uma seção completa neste documento, devido à complexidade desses diretórios.

## 3.3 – Opções específicas.

Os seguintes diretórios, ou links simbólicos para diretórios, devem estar no / se os subsistemas correspondentes estão instalados.

| Diretório | Descrição |
|---|---|
| home | Diretório de usuários (opcional) |
| lib<qual> | Formato alternativo de bibliotecas essenciais compartilhadas (opcional) |
| root | Diretório para o usuário root (opcional) |

Cada diretório listado acima é especificado em pormenor nas subseções abaixo.

## 3.4 – / bin – Comandos binários essenciais de usuários.

## 3.4.1 – Propósito.

/Bin contem comandos que podem ser utilizados tanto pelo administrador do sistemas como pelos outros usuários, mas que são necessários quando não há outro sistema de arquivos montados (por exemplo no modo usuário único). Pode também conter comandos que são utilizados indiretamente por scripts ou Shell.[1]

1. Comandos binários que não são essenciais o suficiente para sejam colocados em / bin, devem ser colocados em / usr / bin, em vez disso. Os itens que são necessários apenas por usuários não-raiz (o X Window System chsh, etc) geralmente não são essenciais
   o suficiente para ser colocado dentro da partição raiz.





## 3.4.2 – Requisitos.

Não deve haver subdiretórios no /bin

Os comandos a seguir, ou links simbólicos para os comandos, são necessários em / bin.

| Comando | Descrição |
|---|---|
| **Cat** | Utilitário para concatenar arquivos de saída standard |
| **chgrp** | Utilitário para mudar as propriedades de um grupo de arquivos |
| **chmod** | Utilitário para mudar as permissões de acesso à um arquivo |
| **chown** | Utilitário para mudar o dono(proprietário) de um arquivo e grupo |
| **cp** | Utilitário para copiar arquivos e diretórios |
| **date** | Utilitário para mostrar ou alterar a data e hora do sistema |
| **dd** | Utilitário para converter e copiar um arquivo |
| **df** | Utilitário para imprimir o espaço utilizado pelo sistema de arquivos de disco |
| **dmesg** | Utilitárop para miprimir ou controlar o buffer de mensagens do kernel (núcleo) |
| **echo** | Utilitário para mostrar na tela uma linha de texto |
| **false** | Utilitário para definir que algo não deve ser feito. |
| **hostname** | Utilitário par a mostrar ou alterar o nome do sistema host (nome do computador) |
| **kill** | Utilitário para enviar sinais para processos (ou matar um processo) |
| **ln** | Utilitário para criar links entre dois arquivos |
| **login** | Utilitário para iniciar uma sessão no sistema |

| | |
|---|---|
| **ls** | Utilitário para listar conteúdo de diretório |
| **mkdir** | Utilitário para criar diretório |
| **mknod** | Utilitário para criar blocos ou caracteres especiais em arquivos |
| **more** | Utilitário para paginar através de texto |
| **mount** | Utilitário para montar um sistema de arquivos |
| **mv** | Utilitário para mover ou renomear arquivos |
| **ps** | Utilitário para relatar status de processos |
| **pwd** | Utilitário para imprimir o nome do diretório de trabalho corrente |
| **rm** | Utilitário para remover arquivos ou diretórios |
| **rmdir** | Utilitário para remover diretórios vazios |
| **sed** | Chamada do editor de textos Sed |
| **sh** | Inicia comandos shell |
| **stty** | Utilitário para mudar e imprimis configurações de terminais |
| **su** | Utilitário par a mudar um ID de usuário |
| **sync** | Utilitário para limpar buffers do sistema de arquivos |
| **true** | Utilitário para definir que algo deve ser feito |
| **umount** | Utilitário para desmontar um sistema de arquivos |
| **uname** | Utilitário para imprimir informações do sistema |

Se /bin /sh não é um shell verdadeiro, deve ser uma ligação física ou simbólica para o comando shell real.
Os comandos ***[*** e ***test*** devem ser colocados juntos em um ou outro /bin ou / usr/bin





## Fundamentação.

Por exemplo bash se comporta diferentemente quando chamado como **sh** ou **bash**. O uso de uma ligação simbólica permite também os usuários facilmente ver que o **/bin/sh** não é um shell verdadeiro.

O requisito paraque o os comandos *[* e ***test*** devam ser incluídos como binários (mesmo que implementado internamente por um shell) é compartilhada com a norma POSIX.2.

## 3.4.3 - Opções Específicas.

Os seguintes programas ou links simbólicos para programas, devem estar no **/bin** se os subsistemas correspondentes estiverem instalados:

| Comando | Descrição |
|---|---|
| **csh** | O Shell C (opcional) |
| **ed** | O editor ‘ed’ (opcional) |

**tar** O utilitário de arquivos “tar” (opcional)
**cpio** O utilitário de arquivos “cpio” (opcional)
**gzip** O utilitário de compressão GNU (opcional)
**gunzip** O utilitário de descompressão GNU (opcional)
**zcat** O utilitário de descompressão GNU (opcional)
**netstat** O utilitário de estatísticas da rede de trabalho (opcional)
**ping** O utilitário de teste de redes de trabalho ICMP (opcional)

Se os programas **gunzip** e **zcat** existirem, elas devem ser simbólicas ou físico para **gzip**. **/bin/csh** pode ser um link simbólico para **/bin/tcsh** ou **/usr/bin/tcsh.**

### Fundamentação.

Os comandos **tar, zip** e **cpio** devem ter sido adicionado para fazer a possível restauração de um sistema (desde que / esteja intacto).

Inversamente, se nenhuma restauração da partição raiz é esperada, então esses binários podem ser omitidos (por exemplo, uma raiz chip ROM, pode ser montada a partir do / usr através de NFS). Se a restauração de um sistema é planejado através de uma rede , FTP ou TFTP (junto com todo o necessário para obter uma conexão FTP) devem estar disponíveis na partição raiz.

## 3.5 - /boot : Arquivos estáticos e de inicialização.

## 3.5.1 – Propósito.

Este diretório contém todo o necessário para o processo de inicialização, exceto arquivos de configuração que não são necessários no momento de boot e mapas de instalação. Assim, o que é armazenado em / boot são os dados que são usados antes do kernel começar a execução de programas em modo usuário. Isso pode incluir salva de setores master de inicialização e setores de mapeamento de arquivos[2].

---

2. Programas necessários para organizar o gerenciador de inicialização para ser capaz de arrancar um arquivo devem ser colocado em / sbin. Os arquivos de configuração de gerenciadores de inicialização deve ser colocado em / etc. O GRUB lê as configurações do arquivo antes de arrancar, de modo que devem ser colocados em / boot. No entanto, se for um arquivo de configuração, deve estar em / etc A resposta aqui é um link simbólico como /etc/grub/menu.lst –> /boot/menu.lst.





## 3.5.2 - Opções Específicas.

O kernel do sistema operacional deve estar localizado em qualquer um dos dois, / ou / boot[3].

## 3.6 - /dev : Arquivos de dispositivos.

## 3.6.1 – Propósito.

O diretório / dev é o local de arquivos especiais ou dispositivos.

## 3.6.2 - Opções Específicas.

Se é possível que os dispositivos em / dev terão de ser criados manualmente, o / dev deve conter um comando chamado MAKEDEV, o qual pode criar dispositivos quando necessário. Ele também pode conter um MAKEDEV.local para todos os dispositivos locais.

Se necessário, MAKEDEV deve ter disposições para a criação de qualquer dispositivo que pode ser encontrado no sistema, não apenas aqueles que instala uma determinada aplicação.

# 3.7 - /etc : Sistema de configurações específicas do host.

## 3.7.1– Propósito.

A hierarquia / etc contém arquivos de configuração. Um "arquivo de configuração " é um arquivo local usado para controlar a operação de um programa, que deve ser estático e não pode ser um executável binário[4].

.

## 3.7.2 – Requisitos.

Arquivos não binários podem estar localizados no /etc[5].

Os seguintes diretórios ou links simbólicos para os diretórios são necessários no arquivo / etc:

| **Diretório** | **Descrição** |
|---|---|
| opt | Configurações para /opt |
| X11 | Configurações para o sistema X Window system (opcional) |
| sgml | Configurações para SGML (opcional) |
| xml | Configurações para XML (opcional) |

## 3.7.3 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios devem estar em / etc, se o subsistema correspondente estiver instalado:

---

3. Em algumas máquinas i386, pode ser necessário o / boot estar localizado em uma partição separada localizado completamente abaixo do cilindro 1024 do dispositivo de arranque devido a limitações de hardware.
Determinados sistemas MIPS requerem uma partição / boot, que é montado em um sistema de arquivos MS-DOS ou qualquer outro tipo de sistema de arquivos que seja acessível pelo firmware. Isso pode resultar em restrições no que diz respeito a nomes utilizáveis dentro de / boot (somente para sistemas afetados).
4. A configuração dos scripts de comando chamado em tempo de inicialização pode assemelhar-se System V, BSD ou outros modelos. Especificações adicionais neste domínio podem ser adicionados a uma futura versão desta norma.
5. É recomendado que os arquivos sejam armazenados em subdiretórios de / etc, em vez de diretamente no / etc





| Diretório | Descrição |
|---|---|
| Opt | Configuração para /opt |

Os seguintes arquivos ou links simbólicos para os arquivos, deve estar em / etc, se o subsistema correspondente estiver instalado[6]:

| Arquivos | Descrição |
|---|---|
| `csh.login` | Arquivo de inicialização Systemwide para logins shell C (opcional) |
| `exports` | controle de acesso a lista do sistema de arquivos NFS (opcional) |

`fstab` Informações estáticas sobre o sistema de arquivos (opcional)
`ftpusers` Controle de acesso a lista de usuários FTP daemon (opcional)
`gateways` Arquivo com a lista de gateways para roteadores (opcional)
`gettydefs` Speed e Configuração de terminal usados pelo getty (opcional)
`group` Arquivo de grupos de usuários (opcional)
`host.conf` Arquivos de configurações do host (opcional)
`hosts` Informações estáticas sobre nomes de hosts (opcional)
`hosts.allow` Arquivos de acessos do Host para TCP wrappers (opcional)
`hosts.deny` Arquivo de acessos do Host para TMP wrappers (opcional)
`hosts.equiv` Lista de verdades do Host para rlogin, rsh, rcp (opcional)
`hosts.lpd` Lista de verdades do Host para lpd (opcional)
`inetd.conf` Arquivo de configurações para o inetd (opcional)
`inittab` Arquivos de configurações para iniciar o sistema (opcional)
`issue` Arquivo de identificação e mensagens de pré-login (opcional)
`ld.so.conf` Lista de diretórios extras para pesquisa nas bibliotecas compartilhadas (opcional)
`motd` Arquivo de mensagens de Post-login do dia (opcional)
`mtab` Informações dinâmicas sobre o sistema de arquivos (opcional)
`mtools.conf` Arquivo de configurações para mtools (opcional)
`networks` Informações estáticas sobre nomes de redes (opcional)
`passwd` Arquivo de senhas (opcional)
`printcap` Base de dados de impressoras habilitadas (opc€ional)
`profile` Arquivo de inicialização do Systemwide para logins sh Shell (opcional)
`protocols` Listagem de protocolos IP (opcional)
`resolv.conf` Arquivos de configurações (opcional)
`rpc` Listagem de protocolos RPC (opcional)
`securetty` Controle de acesso TTY para login root (opcional)
`services` Nome da porta para serviços de rede (opcional)
`shells` Nome de caminho de login shell valido (opcional)
`syslog.conf` Arquivo de configuração para syslogd (opcional)

mtab não comporta a natureza estática do / etc: é excepção por razões históricas[7].

---

6. Sistemas que usam shadow password suíte, terão arquivos de configuração adicionais / etc (/etc/shadow e outros) e programas em /usr/sbin (**useradd, usermod**, e outros).
7. Em alguns sistemas Linux, este pode ser um link simbólico para **/ proc/mounts**, caso em que este não é exceção necessária.





# 3.7.4 - /etc/opt – Arquivo de configurações para /opt.

## 3.7.4.1 – Propósito.

Arquivo de configurações específicas do Host para adicionar pacotes de software aplicativo deve ser instalado dentro do diretório /etc/opt / <subdir>, onde <subdir> é o nome da subárvore em / opt onde os dados estáticos dos pacotes sãoarmazenado.

## 3.7.4.2 – Requisitos.

Nenhuma estrutura é imposta ao arranjo interno do arquivo / etc/opt/<subdir>.  Se um arquivo de configuração deve residir em um local diferente, para que o pacote ou sistema possa funcionar corretamente, ele pode ser colocado em um local diferente de /etc/opt/<subdir>.

### Fundamentação.

Referencia para a fundamentação do /opt.

## 3.7.5  - /etc/X11 – Configurações para o sistema X Windows (Opcional).

## 3.7.5.1 – Propósito.

**/etc/X11** é o local para todo o exército X11-configuração específica. Este diretório é necessária para permitir o controle local  se o / usr é montado somente para leitura.

## 3.7.5.2 - Opções Específicas.

Os seguintes arquivos ou links simbólicos para os arquivos, devem estar  em / etc/X11 se o subsistema correspondente estiver instalado

| Arquivos | Descrição |
|---|---|
| `Xconfig` | Arquivo de configurações para versões antigas do XFree86 (opcional) |
| `XF86Config` | Arquivo de configurações para XFre86 versões 3 e 4 (opcional) |
| `Xmodmap` | Arquivo de modificações para teclados X11 (opcional) |

Subdiretórios do **/etc/X11** podem incluir aqueles para **xdm** e para todos os outros programas (alguns gestores de janelas, por exemplo) que deles necessitam[8].  Recomendamos que os gestores de janelas com apenas um arquivo de configuração que é um padrão ter  arquivos  **.*wmrc** deve ter nome  **sistema.*wmrc** (a menos que haja um nome alternativo amplamente aceito), e não usem  um subdiretório. Qualquer subdiretórios gerenciador de janelas deve ser idênticamente nomeado para oatual gerenciador de janela binário.

8. **/etc/X11/xdm** detém os arquivos de configuração **xdm**. Estes são a maioria dos arquivos encontrados previamente em **/usr/lib/X11/xdm**. Alguns dados variáveis  locais para o **xdm** são armazenado em **/var/lib/xdm**.





## 3.7.6  - /etc/sgml – Arquivo de configurações para SGML (opcional).

## 3.7.6.1 – Propósito.

Arquivos de configurações genéricas que define parâmetros de alto nível dos sistemas SGML estão instalados aqui. Arquivos com ***. conf** indicam nomes de arquivos de configuração genérica. Arquivos com nomes ***. cat** são de catálogos centralizados específicos DTD, contendo referências de todos os outros catálogos necessários para a utilização do dado DTD. O arquivo de super catálogo ***catalog*** tem referências do todos os catálogos centralizado.

## 3.7.7 - /etc/xml – Arquivo de configurações para XML (opcional).

## 3.7.7.1 – Propósito.

Arquivos genéricos de configuração que definem parâmetros de alto nível dos sistemas XML estão instaladas aqui. Arquivos com
***.conf** indicam nomes de arquivos de configuração genérica. O Arquivo de super catálogo ***catalog*** tem as referências centralizadas de todos os catálogos.

## 3.8 - /home : Diretório inicial de usuário (opcional).

## 3.8.1 – Propósito.

/ home é um conceito bastante normal, mas é claramente um sistema de arquivos específicos [9]. A instalação será diferente de host para host. Portanto, nenhum programa deve basear-se neste local [10.]

## 3.8.2 – Requisitos.

Arquivos de configuração de usuário para aplicações específicas são armazenados no diretório home do usuário em um arquivo que começa com caracter “.” (um ponto).. Se um aplicativo precisa criar mais de um arquivo ponto, então eles devem ser colocados em um subdiretório com o nome iniciando com um caracter “.” (um "diretório ponto"). Neste caso, os arquivos de configuração não devem começar com o caracter “.” [11].

## 3.9 - /lib : Bibliotecas essenciais compartilhadas e módulos Kernel.

## 3.9.1 – Propósito.

O diretório **/lib** contém aquelas imagens de bibliotecas compartilhadas necessárias para iniciar o sistema e executar os comandos no sistema de arquivos raiz, isto é, binários em **/bin** e **/sbin** [12]**.**

---

9. Diferentes pessoas preferem colocar as contas de usuário em uma variedade de lugares. Esta seção descreve apenas uma sugestão de colocação para o diretório home do usuário, no entanto, recomendamos que todas as distribuições FHS compatíveis usem esta como a localização padrão para diretórios de inicialização de usuários.
   Em sistemas pequenos, cada diretório de usuário é tipicamente um dos subdiretórios de **/home** como **/home/smith, /home/Torvalds**, **/home/operador**, etc Em sistemas grande (especialmente quando os diretórios /home são compartilhados entre muitos hosts usando NFS) é útil subdividir os diretórios home de usuário. Subdivisões podem ser obtidas utilizando-se subdiretórios tal como /home/pessoal , / home/convidados, /home/alunos, etc
10. Se você quiser procurar um usuário fora do diretório home, você deve usar a biblioteca de funções de procura ***getpwent*** (3) do /etc passwd, porque as informações do usuário podem estar armazenados remotamente usando sistemas como NIS.
11. Recomenda-se que, além da gravação automática de programas e arquivos de bloqueio, devem abster-se de criar arquivos ou diretórios de pontos em um diretório /home sem intervenção do usuário.
12. Bibliotecas compartilhadas que são necessárias apenas para binários no diretório usr (como qualquer binários X Window), não deve estar em **/lib**. Somente as bibliotecas compartilhadas necessárias para executar binários em ***/bin*** e ***/sbin*** pode estar aqui. Em particular, a biblioteca ***libm.so.****também pode ser colocado em **/usr/lib** se não é exigido por qualquer coisa em /bin ou /sbin.

### 3.9.2 – Requisitos.

Pelo menos um de cada um dos seguintes nomes de arquivos padrões são necessários (que podem ser arquivos ou links simbólicos).

| Arquivo | Descrição |
|---|---|
| `libc.so.*` | A biblioteca C de linkar dinamicamente (opcional) |
| `ld*` | O tempo de execução para disparar a linkedição (opcional) |

Se um pré-processador C é instalado, ***/lib/cpp*** deve ser uma referência a ele, por razões históricas [13].

### 3.9.3 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar em **/lib**, se o subsistema correspondente estiver instalado:

| Diretório | Descrição |
|---|---|
| `modules` | Módulos Kernel executáveis (opcional) |

## 3.10 - /lib<qual> : Bibliotecas essenciais compartilhadas formatos alternativos (opcional).

### 3.10.1 – Propósito.

Pode haver uma ou mais variantes do diretório ***/lib*** em sistemas que suportam mais de um formato binário e que requerem bibliotecas separadas [14].

### 3.10.2 – Requisitos.

Se um ou mais desses diretórios existem, os requisitos para o seu conteúdo é o mesmo que o diretório normal */lib* , exceto que ***/lib<qual>/cpp*** não é necessário [15].

## 3.11 - /media : Ponto de montagem para mídias removíveis.

### 3.11.1 – Propósito.

Este diretório contém os subdiretórios que são usados como pontos de montagem para mídias removíveis, como disquetes discos, CDROMs e discos ZIP.

13. A colocação de costume deste binário é em ***/usr/bin/cpp***.
14. Isto é comumente usado em sistemas de 64-bit ou 32-bit que suportam múltiplos formatos binários, mas requerem

bibliotecas do mesmo nome. Neste caso*,* ***/lib32*** e ***/lib64*** poderiam ser os diretórios de bibliotecas e ***/lib*** uma ligação simbólica para um deles.

15. ***/lib<qual>/cpp*** ainda é permitido: isto é permitido nos casos em que /lib e /lib<qual> são os mesmos (um é um link simbólico para o outro).





## Fundamentação.

Historicamente tem havido uma série de outros lugares diferentes usados para a montagem de mídia removível, como **/cdrom**, **/mnt** ou **/mnt/cdrom**. Colocar os pontos de montagem para todas as mídias removíveis diretamente no diretório raiz potencialmente resultar em um grande número de diretórios extra no /. Embora o uso de subdiretórios em **/mnt** como um ponto de montagem foi recentemente comum, ele entra em conflito com uma tradição muito mais antiga do uso do **/mnt** diretamente como um ponto de montagem temporário.

### 3.11.2 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar em /media, se o subsistema correspondente estiver instalado.

| Diretorio | Descrição |
|---|---|
| `floppy` | Dispositivo de Floppy (opcional) |
| `cdrom` | Dispositivo de CD-ROM (opcional) |
| `cdrecorder` | Gravador de CD (opcional) |
| `zip` | Dispositivo de ZIP (opcional) |

Em sistemas onde mais de um dispositivo de montagem, existe para montar um certo tipo de mídia, diretórios podem ser criado acrescentando um dígito ao nome daqueles disponíveis acima começando " 0", mas o nome não qualificado também deve existir [16].

## 3.12 - /mnt : Ponto de montagem para um sistema de arquivos montado temporariamente.

### 3.12.1 – Propósito.

Este diretório é fornecido para que o administrador do sistema possa, temporariamente, montar um sistema de arquivos, conforme necessário. O conteúdo deste diretório é uma questão local e não deve afetar a maneira pela qual todo o programa é executado.

Este diretório não deve ser utilizado por programas de instalação: um diretório temporário adequado não está em uso pelo sistema quando deve ser utilizado.

## 3.13 - /opt : Adição de pacotes de softwares aplicativos.

### 3.13.1 – Propósito.

**/opt** é reservado para a instalação de add-on (adição) pacotes de software aplicativo.

Um pacote para ser instalado em **/opt** deve localizar seus arquivos estáticos em uma árvore de diretórios /opt /<package> ou /opt/<provider> separado, onde <package> é um nome que descreve o pacote de software e <provider> é o fornecedor de nome registrados LANANA.

16. Uma implementação compatível com duas unidades de CD-ROM pode ter **/media/cdrom0** e **/media/cdrom1** com **/media/ cdrom** sendo um link simbólica para um desses.





## 3.13.2 – Requisitos.

| Diretorio | Descrição |
|---|---|
| <package> | Pacotes de objetos estáticos |
| <provider> | LANANA nome do fornecedor registrado |

Os diretórios **/opt /bin, /opt/doc, /opt/include, /opt info, /opt/lib** e **/opt/man** são reservados para uso local do administrador do sistema. Os pacotes podem fornecer arquivos de "front-end" destinados a ser colocados em (para links ou cópias) nesses diretórios reservados pelo administrador do sistema local, mas devem funcionar normalmente na ausência desses diretórios reservados.

Programas para serem invocada pelo usuário devem estar localizado no diretório /opt /<package>/bin ou sob a hierarquia **opt/<provider>** . Se o pacote inclui páginas de manual Unix, eles devem estar localizados em **/opt/<package>/share/man** ou sob a hierarquia **/opt/<provider>**, e a mesma subestrutura **/usr/share/man** deve ser usado.
Pacote de arquivos que são variáveis (mudam no funcionamento normal) devem ser instalado em **/var/opt**. Veja a seção sobre **/var/opt** para mais informações.

Arquivos específicos de configurações de Host devem ser instalados em **/etc/opt**. Veja a seção sobre **/etc** para mais informações.

Não devem existir outros pacotes de arquivos fora das hierarquias **/opt, /var/opt** e **etc/opt**, exceto para aqueles pacotes de arquivos que deve residir em locais específicos dentro da árvore de arquivos, a fim de funcionar corretamente. Por exemplo, os arquivos de bloqueio de dispositivo devem ser colocado em **/var/lock** e dispositivos devem estar localizados em **/dev**.

Distribuições pode instalar o software em **/opt**, mas não deve alterar ou apagar o software instalado pelo administrador de sistema local sem o parecer favorável do administrador do sistema local .

## Fundamentação.

O uso de **/opt** pra adicionar um software é uma prática bem estabelecida na comunidade UNIX. O System V Interface de Aplicação Binária [AT&T 1990], baseado no System V Interface Definition (terceira edição), prevê um **/opt** estrutura muito semelhante ao aqui definido.

A Intel Binary Compatibility Standard v. 2 (iBCS2) também fornece uma estrutura semelhante à do **/opt**.

Geralmente, todos os dados necessários para suportar um pacote em um sistema deve estar presente em **/opt/<package>,** incluindo arquivos destinados a ser copiado para **/etc/opt/ <package>** e **/var/opt/<package>** bem como diretórios reservados em **/opt**.

As restrições mínimas sobre a distribuição usando **/opt** são necessárias porque os conflitos são possíveis entre distribuições instaladas e software instalado localmente, especialmente no caso dos caminhos fixos encontrados em alguns software binário.

A estrutura de diretórios abaixo de **/opt/<provider>** é deixada para o acondicionador do software, embora recomenda-se que pacotes devam ser instalados em /opt/<provider>/<package> e seguem uma estrutura similar com as orientações para o pacote **/opt/package**. Uma razão válida para afastando-se esta estrutura é o suporte para pacotes que possam ter arquivos instalados em **/opt/<provider>/lib** ou **/opt/<provider>/bin**.

# 3.14 - /root : Diretório residente para o usuário root (opcional).

## 3.14.1 – Propósito.

O diretório residente da conta de root pode ser determinado pelo desenvolvedor ou preferência local, mas este é o local padrão recomendado [17].

# 3.15 - /sbin : Sistemas binários.

## 3.15.1 – Propósito.

Utilitários usados para a administração do sistema (outros comandos de root apenas) são armazenados em **/sbin, usr/ sbin**, e **/usr/local/sbin**. **/sbin** contém binários essenciais para a inicialização do sistema (boot), restauração, recuperação e/ou reparação dos sistema, em conjunto com os binários em **/bin** [18]. Programas reconhecidos e executados no **/usr** após ser montado (quando não há problemas) são geralmente colocados em **/usr/sbin**. Localmente programas de administração do sistema instalado deve ser colocado em **/usr/local/sbin** [19]**.**

## 3.15.2 – Requisitos.

Os comandos a seguir, ou links simbólicos para os comandos, são necessários em **/sbin**.

| Comando | Descrição |
|---|---|
| **shutdown** | Comando para derrubar o sistema(colocar para baixo). |

## 3.15.3- Opções Específicas.

Os seguintes arquivos ou links simbólicos para os arquivos, deve estar em **/sbin**, se o subsistema correspondente estiver instalado:

| Comando | Descrição |
|---|---|
| `fastboot` | Reinicialização rápida do sistema sem checar os discos (opcional) |
| `fasthalt` | Parada rápida do sistema sem checar os discos (opcional) |
| `fdisk` | Tabela de manipulação de partições de discos (opcional) |
| `fsck` | Utilitário para checar e reparar o sistema de arquivos (opcional) |
| `fsck.*` | Utilitário para checar e reparar o sistema de arquivos, para um sistema de arquivos específico (opcional) |
| `getty` | O programa getty (opcional) |
| `halt` | Comando para parada do sistema (opcional) |
| `ifconfig` | Interface de configuração de rede (opcional) |
| `init` | Processo inicial (opcional) |

| `mkfs` | Comando para criar um sistema de arquivos (opcional) |
|---|---|
| `mkfs.*` | Comando para criar um sistema de arquivos específico (opcional) |
| `mkswap` | Comando para criar uma área de swap (opcional) |
| `reboot` | Comando para reinicializar o sistema (opcional) |
| `route` | Utilitário para tabela de roteamento IP (opcional) |
| `swapon` | Habilitar paginação e swap (opcional) |
| `swapoff` | Desabilitar paginação e swap (opcional) |
| `update` | Daemon para limpar periodicamente os buffers do sistema de arquivos (opcional) |

---

17. Se o diretório home do usuário root não está armazenado na partição raiz, será necessário fazer certamente um padrão para /, senão ele não poderá ser localizado.
    Não é recomendável o uso da conta root para tarefas que podem ser executados como um usuário sem privilégios, e que seja utilizado exclusivamente para a administração do sistema. Por este motivo, recomendamos que subdiretórios para mail e outras aplicações não aparecem no diretório home da conta de root, e que os papéis escritos para a administração, como root, postmaster, e webmaster sejam enviados para os usuário apropriados.

18. Originalmente, binários /sbin binários foram mantidos em /etc.





19. Decidir que coisas vão dentro do diretório**"sbin**" é simples: se um usuário normal (não um administrador do sistema) executar algo constantemente, então isso deve ser colocado em um dos diretórios "bin". Os usuários comuns não devem ter qualquer referencia aos diretórios **/bin** em seus paths (caminhos).

    Por exemplo, arquivos como **chfn** que os usuários só usam ocasionalmente devem ainda ser colocados em **/usr/bin**. **ping**, embora seja absolutamente necessário para o root (recuperação de rede e diagnóstico) é freqüentemente usada por usuários e deve estar em **/bin** por várias razões.

    Nós recomendamos que os usuários tenham permissão de leitura e execução para tudo no **/sbin** exceto, talvez, determinados programas como **setuid** e **setgid**. A divisão entre **/bin** e **/sbin** não foi criado por razões de segurança ou para impedir que usuários vejam o sistema operacional, mas para proporcionar uma boa partição entre os binários de uso geral e os que são usado principalmente para tarefas de administração. Não há nenhuma vantagem em fazer a segurança inerente ao **/sbin** fora dos limites para os usuários.

---

# 3.16 - /srv : Dados para serviços providos por esse sistemas.

## 3.16.1 – Propósito.

/srv contém dados específicos do que é servido por este sistema.

### Fundamentação.

O objetivo principal de especificar isso assim é que os usuários podem encontrar a localização dos arquivos de dados pra serviços particulares (específicos) e também para os serviços que exigem uma única árvore para os dados de leitura, escrita e scripts de dados (como scripts CGI) pode ser razoavelmente colocado. Os dados que são apenas de interesse específico para um determinado usuário deve ir no diretório home do usuário.

A metodologia utilizada para nomear subdiretórios do **/srv** não é especificada e atualmente não há consenso sobre como isso deve ser feito. Um método para estruturação de dados em **/srv** é por protocolo, por exemplo. **ftp, rsync, www**, **e cvs**. Em sistemas grandes, pode ser útil estruturar /srv pelo contexto administrativo, como **/srv/physics/www**, **/srv/compsci/cvs**, etc Esta configuração será diferente de um hospedeiro para outro. Portanto, nenhum programa deve contar com uma estrutura específica do subdiretório **/srv** existente ou de dados, necessariamente, armazenado em **/srv**. Entretanto se **/srv** existir deve sempre existir em sistemas FHS compatível e deve ser usado como o local padrão para esses dados.

Distribuições deve tomar cuidado para não remover os arquivos localmente colocados nestes diretórios sem permissão do administrador [20].

## 3.17 - /tmp : Arquivos temporários.

## 3.16.1 – Propósito.

O diretório **/tmp** devem ser disponibilizados para programas que exigem arquivos temporários.

Os programas não devem assumir que qualquer arquivo ou diretório em /tmp são preservados apenas durante chamadas de programas.

---

20. Isto é particularmente importante uma vez que estas áreas, muitas vezes contêm arquivos instalado inicialmente pelo distribuidor e adicionados pelo administrador.





### Fundamentação.

padrão IEEE P1003.2 (POSIX, parte 2) faz exigências que são semelhantes para a seção acima.

Embora os dados armazenados no **/tmp** pode ser eliminada em um local específico, é recomendável que os arquivos e diretórios localizados em **/tmp** sejam excluída sempre que o sistema é iniciado.

FHS adicionou esta recomendação com base em precedentes históricos e de prática comum, mas não torna isso uma exigência, porque a administração do sistema não está dentro do âmbito de aplicação desta norma.

# Capítulo 4. – A hierarquia /usr

## 4.1 – Propósito.

***/usr** é* a segunda parte mais importante do sistema de arquivos. ***/usr*** é compartilhável, para dados somente leitura. Isso significa que **/usr** pode ser compartilhado entre várias hosts FHS compatíveis, e não deve ser para escrita. Qualquer informação que seja Específica de um hospedeiro ou varia com o tempo deve ser armazenado em outro lugar.

Grandes pacotes de software não devem usar subdiretórios diretamente dentro da hierarquia **/usr**.

## 4.2 – Requisitos.

Os seguintes diretórios ou links simbólicos para diretórios, são necessários em **/usr**

| **Diretório** | **Descrição** |
|---|---|
| `bin` | A maioria dos comandos de usuários |
| `include` | Header de arquivos incluídos por programas C |
| `lib` | Bibliotecas |
| `local` | Hierarquia local (vazio após a maioria das instalações) |
| `sbin` | Sistemas binários não vitais |
| `share` | Dados de arquitetura independente (compartilhados/públicos) |

## 4.3- Opções Específicas.

| **Diretório** | **Descrição** |
|---|---|
| `X11R6` | Sistema XWindow, versão 11 release 6 (opcional) |
| `games` | Binários de jogos e educacionais (opcional) |
| `lib<qual>` | Biblioteca de formatos alternativos (opcional) |

src Códigos fontes (opcional)

Uma exceção é feita para o sistema X Window por causa de consideráveis precedente e importante prática amplamente aceita

Os seguintes links simbólicos a diretórios podem estar presentes. Essa possibilidade é baseada na necessidade de preservar a compatibilidade com sistemas mais antigos, até que todas as implementações possam ser consideradas como uso da hierarquia **/var.**

/usr/spool -> /var/spool

/usr/tmp -> /var/tmp

/usr/spool/locks -> /var/lock

Uma vez que um sistema não exige qualquer um dos links simbólicos acima, o link pode ser removido, se desejar.





# 4.4 – /usr/X11R6 – Sistema X Window, Versão 11 Release 6 (opcional).

## 4.4.1 – Propósito.

Esta hierarquia é reservada para o sistema X Window, versão 11 release 6 e arquivos relacionados.

Para tornar mais simples e fazer o XFree86 mais compatível com o sistema X Window em outros sistemas, os seguintes links simbólicos devem estar presentes se **/usr/X11R6** existir:

/usr/bin/X11 -> usr/X11R6/bin /

/usr/lib/X11 -> / usr/X11R6/lib/X11

/usr/include/X11 -> / usr/X11R6/include/X11

Em geral, o software não deve ser instalado ou geridos através dos links simbólicos acima. São destinados a utilização pelos usuários apenas. A dificuldade está relacionada com a versão do sistema X Window em períodos de transição, é impossível saber qual a versão do X11 está em uso.

## 4.4.2 - Opções Específicas.

Dados de Host específicos em **/usr/X11R6/lib/X11** deve ser interpretada como um arquivo de demonstração. Aplicações requerendo informações sobre o host atual devem fazer referência à arquivo de configuração em **/etc/X11**, que pode ser vinculado a um arquivo em **/usr/X11R6/lib** [1].

# 4.5 – /usr/bin – A maioria dos comandos de usuário.

## 4.5.1 – Propósito.

Este é o diretório primário de comandos executáveis do sistema.

## 4.5.2 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios devem esta em **/usr/bin**, se o subsistema correspondente estiver instalado.

| **Diretório** | **Descrição** |
|---|---|
| mh | Comandos para o sistema de tratamento do correio MH (opcional) |

/usr/bin/X11 deve ser um lim simbólico para to /usr/X11R6/bin se o último existir.

Os seguintes arquivos , ou links simbólicos para arquivos, deve estar em /usr/bin, se o subsistema correspondente estiver instalado:

| **Comando** | **Descrição** |
|---|---|
| **Perl** | A linguagem de extração prática e relatório (opcional) |
| **Python** | A linguagem interpretada Python (opcional) |
| **tclsh** | Shell simples contendo o interpretador Tcl (opcional) |
| **wish** | Shell de moldura simples de janela Tcl/Tk (opcional) |
| **expect** | Programa para diálogo interativo (opcional) |

1. Exemplos de arquivos de configuração, incluem Xconfig, XF86Config, ou system.twmrc





### Fundamentação.

Porque interpretadores de script shell (invocados com #! <caminho> Na primeira linha de um script de shell) não podem contar com um caminho, é vantajoso para padronizar suas localizações. Os interpretadores Bourne-shell e C-shell já estão fixos em **/bin**, mas Perl, Python e Tcl são freqüentemente encontrados em muitos lugares diferentes. Eles podem ser links simbólicos para a localização física dos interpretadores shell.

# 4.6 – /usr/include – Diretórios para inclusão padrão de arquivos.

## 4.6.1 – Propósito.

Este é o lugar onde todo o sistema de uso geral, incluindo arquivos da linguagem de programação C devem ser colocados.

## 4.6.2 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar no diretório **/usr/include**, se o correspondente subsistema estiver instalado:

| **Diretório** | **Descrição** |
|---|---|
| bsd | Inclusão de arquivos com compatibilidade BSD (opcional) |

O link simbólico /usr/include/X11 deve apontar para /usr/X11R6/include/X11 se o último existir.

## 4.7 – /usr/lib – Bibliotecas de programas e pacotes.

### 4.7.1 – Propósito.

**/ usr/lib** inclui arquivos objetos, bibliotecas e binários internos que não se destinem a ser executados diretamente pelos usuários ou shell scripts [2].

Os aplicativos podem usar um único subdiretório em **/usr/lib**. Se um aplicativo usa um subdiretório, todos os dados dependem exclusivamente da arquitetura utilizada pela aplicação e deve ser colocado dentro desse subdiretório [3].

### 4.7.2 - Opções Específicas.

Por razões históricas, **/usr/lib/sendmail** deve ser um link simbólico para **/usr/sbin/sendmail**, se este existir [4].

---

2. Arquiteturas independentes diversas, arquivos estáticos de aplicações específicas e subdiretórios devem ser colocados em **/usr/share**.
3. Por exemplo, o subdiretório perl5 para as bibliotecas e módulos do perl5.
4. Alguns comandos executáveis tais como sendmail makewhatis também têm sido tradicionalmente colocados na **/usr/lib**. makewhatis é um binário interno e deve ser colocado em um diretório binário, apenas acessos de usuários **catman**. Binários recentes sendmail são colocados por padrão no diretório **/usr/sbin**. Além disso, os sistemas que utilizam um agente sendmail de transferência compatível devem fornecer /usr/sbin/sendmail como um link simbólico para o executável apropriado.





Se /lib/X11 existir, /usr/lib/X11 deve ser um link simbólico para /lib/X11, ou para qualquer /lib/X11 é um link simbólico [5].

## 4.8 – /usr/lib<qual> – Bibliotecas de formatos alternativos (opcional).

### 4.8.1 – Propósito.

**/usr/lib<qual>** desempenha o mesmo papel, que **/usr/lib** para um formato alternativo binário, exceto que o link simbólico **/usr/lib<qual>/sendmail** e **/usr/lib<qual>/X11** não são necessários [6].

### 4.8.2 – /usr/local - Hierarquia local.

#### 4.8.2.1 – Propósito.

A hierarquia /usr/local é para ser usado pelo administrador do sistema durante a instalação do software localmente. É preciso estar seguro do software ser substituído quando o sistema é atualizado. Ele pode ser usado para programas e dados que são compartilháveis entre um grupo de hosts, mas não encontrado no diretório /usr.

Software instalado localmente deve ser colocado em /usr/local ao invés de /usr, a menos que ele esteja sendo instalado para substituir ou atualizar um software em /usr [7].

## 4.8.2.2 – Requisitos.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar no diretório /usr/local.

| Diretório | Descrição |
|---|---|
| bin | Binários locais |
| etc | Configurações do sistema para host específicos para binários locais |
| games | Binários locais para jogos |
| include | Header de arquivos C locais |
| lib | Bibliotecas locais |
| man | Manuais on-line locais |
| sbin | Binários do sistema local |
| share | Hierarquia de arquitetura independente local (compartilhamento) |
| src | Códigos fontes locais |

Em nenhum outro diretório, exceto aqueles listados abaixo, deve estar em **/usr/local** o sistema FHS compatível após a primeira instalação do sistema.

## 4.8.2.3 - Opções Específicas.

Se diretórios **/lib/<qual>** ou **/usr/lib<qual>** existirem, os diretórios equivalentes também devem existirem no **/usr/local**.

**/usr/local/etc** pode ser um link simbólico para **/etc/local.**

---

5. Dados específicos de Host para o sistema X Window não devem ser armazenados em **/usr/lib/X11**.Arquivos de configurações específicas de Host, como Xconfig ou XF86Config devem ser armazenados em **/etc/X11**. Isto inclui dados de configuração, como **system.twmrc** mesmo que só seja feita uma ligação simbólica para um arquivo de configuração mais global (provavelmente em **/usr/X11R6/lib/X11**).
6. O caso onde **/usr/lib** e **/usr/lib<qual>** são os mesmos (um é um link simbólico para o outro), esses arquivos e os subdiretórios por aplicação, deverão existir.
7. Software colocados em **/** ou **/usr** pode ser sobrescrito por atualizações do sistema (embora recomendamos para distribuições não substituir os dados no arquivo **/etc** nestas circunstâncias). Por este motivo, o software local não deve ser colocado fora de /usr/local sem uma boa razão.





### Fundamentação.

A consistência da pasta **/usr/local/etc** é benéfica para os instaladores, e já é utilizado em outros sistemas. Como todos, **/usr/local** precisa ser apoiada para reproduzir um sistema, ele não introduz sobrecarga de manutenções adicionais, mas um link simbólico para /etc/local é adequado se todos os sistemas procurados tiverem uma configuração sob uma hierarquia.

Note que /usr/etc ainda não é permitido: programas em /usr devem colocar os arquivos de configuração em /etc

## 4.9 – /usr/local/share.

Os requisitos para o conteúdo deste diretório é o mesmo para **/usr/share**. A única restrição adicional é que os diretórios **/usr/local/share/man** e **/usr/local/man** devem ser sinônimos (em geral, isso significa que um deles deve ser um link simbólico)[8].

## 4.10 – /usr/sbin– Sistemas binários padrões não essenciais.

### 4.10.1 – Propósito.

Este diretório contém os binários não-essenciais utilizados exclusivamente pelo administrador do sistema. Programas de administração do sistema que são necessários para a reparação do sistema, a recuperação do sistema de montagem /usr ou outras funções essenciais devem ser colocados em /sbin em vez disso [9].

## 4.11 – /usr/share – Dados de arquiteturas dependentes (compartilhadas).

### 4.11.1 – Propósito.

A hierarquia **/usr/share** é para todos as arquiteturas independentes, arquivos de dados só de leitura [10].

Essa hierarquia é destinado a ser partilhado entre todas as plataformas de arquitetura de um sistema operativo dado, assim, por exemplo, um site com i386, Alpha, e plataformas PPC pode manter um único diretório **/usr/share** que é montado centralmente. Note, entretanto, que o **/usr/share** não é normalmente destinada a ser partilhada por outros sistemas operacionais diferentes ou por diferentes versões do mesmo SO.

Qualquer programa ou pacote que contém ou requer dados que não precisam ser modificados devem armazenar os dados em **/usr/share** (ou **/usr/local/share**, se instalado localmente). Recomenda-se que um subdiretório deve ser utilizado em **/usr/share** para esta finalidade.

Dados de jogos armazenados em **/usr/share/games** devem ser dados puramente estático. Quaisquer arquivos modificáveis, como arquivos de pontuação, logs de jogos, e assim por diante, devem ser colocados em **/var/games.**

### 4.11.2 – Requisitos.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar em **/usr/share**

---

8. **/usr/local/man** pode ser preterido em futuras versões FHS, por isso, se tudo for igual, fazer um link simbólico parece sensato.
9. Programas de administração do sistema instalado localmente deve ser colocado em **/usr/local/sbin**.
10. Grande parte destes dados originalmente colocados em **/usr (man, doc)** ou **/usr/lib(dict, terminfo, zoneinfo).**





| **Diretório** | **Descrição** |
|---|---|
| `man` | Manual on-line |
| `misc` | Dados de arquiteturas independentes diversas |

### 4.11.3 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar em **/usr/share**, se o subsistema correspondente estiver instalado:

| Diretório | Descrição |
|---|---|
| dict | Lista de palavras (opcional) |
| doc | Documentações diversas (opcional) |
| games | Arquivos de dados estáticos para /usr/games (opcional) |
| info | Diretório primário do sistema de informações GNU (opcional) |
| locale | Informações locais (opcional) |
| nls | Catálogo de mensagens para suporte de linguagem nativa (opcional) |
| sgml | Dados SGML (opcional) |
| terminfo | Diretórios para base da dados terminfo (opcional) |
| tmac | Macros troff não distribuídas com groff (opcional) |
| xml | Dados XML (opcional) |
| zoneinfo | Informações e configurações Timezone (opcional) |

Recomenda-se que aplicações específicas, de diretórios de arquiteturas independentes sejam colocados aqui. Esses diretórios incluem **groff, Perl, ghostscript, texmf** e **kbd** (Linux) ou **syscons** (BSD). Eles podem, no entanto, serem colocados em **/usr/lib** para compatibilidade, de discrição do distribuidor. Da mesma forma, uma hierarquia /usr/lib/games pode ser utilizada para além da hierarquia /usr/share/games se o distribuidor deseje colocar alguns dados do jogo lá.

## 4.11.4 - /usr/share/dict – Listas de palavras (opcional).

## 4.11.4.1 – Propósito.

Este diretório é o local das listas de palavras sobre o sistema; Tradicionalmente este diretório contém apenas o arquivo de palavras inglesas, que é usado pelo **look** (1) e vários programas de ortografia. Palavras podem usar tanto ortografia americana como britânica.

### Fundamentação.

A razão para que listas de palavras só são encontradas aqui é que elas são os únicos arquivos comum a todos os corretores ortográficos.





## 4.11.4.2 - Opções Específicas.

Os seguintes arquivos ou links simbólicos para os arquivos, deve estar em **/usr/share/dict**, se o subsistema correspondente estiver instalado:

| **Arquivo** | **Descrição** |
| --- | --- |
| `words` | Lista de palavras inglesas (opcional) |

Sites que exigem tanto a ortografia americana quanto a britânica pode apontar palavras para **/usr/share/dict/american-Inglês** ou **/usr/share/dict/British-Inglês**.

Listas de palavras para outros idiomas podem ser adicionados usando o nome em Inglês para o idioma, por exemplo, **/usr/share/dict/francês, /usr/share/dict/dinamarquês**, etc Estes devem, se possível, usar um conjunto de caracteres ISO 8859 que é apropriado para o idioma em questão e, se possível, o conjunto de caracteres Latin1 (ISO 8859-1) deve ser utilizado (isto muitas vezes não é possível).

Outras listas de palavras devem ser incluídas aqui, se estiverem presente.

## 4.11.5 - /usr/share/man – Paginas de manual.

## 4.11.5.1 – Propósito.

Esta seção detalha a organização para páginas de manual em todo o sistema, incluindo **/usr/share/man**. Além disso consulte a seção sobre /var/cache/man.

O principal **<mandir>** do sistema é **/usr/share/man. /usr/share/man** contém informações do manual sobre comandos e dados entre **/** e **/ usr** [11].

Páginas de manual são armazenados em **<mandir>/<locale>/man<section>/<arch>.** Uma explicação de **<mandir>, <locale>, <section>** e **<arch>** é dado abaixo.

A seguir uma descrição de cada

- man1: manual de programas de páginas que descrevem os comandos acessíveis ao público estão contidas neste capítulo. A documentação da maioria dos programas que o usuário vai precisar usar está localizado aqui.
- man2: Sistema de chamadas Esta seção descreve todas as chamadas de sistema (pedidos para o kernel para executar operações).
- man3: Biblioteca de funções e subrotinas A seção 3 descreve as rotinas de bibliotecas de programas que não são chamadas diretas aos serviços do kernel. Este capítulo 2 só é realmente de interesse para programadores.
- man4: A seção4 – Arquivos especiais - descreve os arquivos especiais, funções de controladores (driver), e suporte para redes disponíveis no sistema. Normalmente, este dispositivo inclui os arquivos encontrados em **/dev** e a interface do kernel para suporte de protocolo de rede.
- man 5: Formatos de arquivo formatos. Os formatos para muitos arquivos de dados são documentados na seção 5. Isso inclui vários arquivos **include**, arquivos de saída de programas e arquivos de sistema.
- man6: Jogos. Este capítulo traz os documentos de jogos, demos e programas geralmente trivial. Pessoas diferentes têm noções variadas sobre como isso é indispensável.
- man7: Diversos páginas de instruções que são difíceis de classificar são designadas como sendo a secção 7. O troff processamento de texto e outros pacotes de macro são encontrados aqui.
- man8: Programas de administração do sistema usado por administradores de sistema para operação e manutenção do sistema estão documentadas aqui. Alguns desses programas também são ocasionalmente úteis para usuários normais.

11. Obviamente, não há páginas de manual em **/** porque eles não são necessários em tempo de boot nem são realmente necessários em emergências.

---

## 4.11.5.2 - Opções Específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar no diretório **/usr/share/<mandir>/<locale>,** a menos que estejam vazios [12]:

| Diretório | Descrição |
|---|---|
| man1 | Programas de usuário (opcional) |
| man2 | Chamadas de sistema (opcional) |
| man3 | Chamadas de bibliotecas (opcional) |
| man4 | Arquivos especiais (opcional) |
| man5 | Arquivos de formatos (opcional) |
| man6 | Jogos (opcional) |
| man7 | Diversos (opcional) |
| man8 | Administração do sistema (opcional) |

O componente <section> descreve a seção do manuais

Previsões devem ser feitas na estrutura do **/usr/share/man** para apoiar páginas do manual que estão escritas em diferentes línguas (ou múltiplo). Estas disposições devem ter em conta o armazenamento e as referências dessas páginas de manual. Fatores relevantes incluem línguas (incluindo as diferenças geográficas básicas), e configurações de códigos de caracteres.

Esta nomenclatura de subdiretórios linguagens do **/usr/share/man** é baseado no anexo E do POSIX 1003.1 norma que descreve a seqüência de identificação de localidade - o método mais bem aceito para descrever uma cultura de ambiente. A String <locale> é: **<language> [_<territory>] [.<character-set>] [,<version>]**

O campo **<language>** deve seguir as normas do ISO 639 (um código para a representação dos nomes dos idiomas). Ele deve ter o tamanho de dois caractéres e especificadas com letras minúsculas somente.

O campo **<territory>** deve ser o código de duas letras da ISO 3166 (especificação de uma das representações de países), se possível. (A maioria das pessoas estão familiarizadas com os códigos de duas letras usadas para os códigos de país em um endereço de e-mail). Deve ter dois caracteres de tamanho e especificadas com letras maiúsculas apenas [13].

O campo **<character-set>** deve representar o padrão descrevendo o conjunto de caracteres. Se o campo <character-set> for apenas uma especificação numérica, o número representa o número do padrão internacional, descrevendo o conjunto de caracteres. Recomenda-se que esta seja uma representação numérica, se possível (especialmente norma ISO), não incluir símbolos de pontuação adicional, e que qualquer letra seja em minúsculas.

Um parâmetro especificando uma versão **<version>** do perfil deve ser colocada depois do campo **<character-set>,** delimitada por uma vírgula. Isso pode ser usado para destacar diferentes necessidades culturais, por exemplo, a ordem do dicionário versus a a ordem de um sistemas de agrupamento orientado. Esta norma recomenda não usar o campo <version>, a menos que seja estritamente necessário.

Sistemas que utilizam uma linguagem única e um código estabelecido para todas as páginas do manual podem omitir a subcadeia **<locale>** e armazenar todas as páginas de manual em **<mandir>.** Por exemplo, sistemas que só tem páginas de manual Inglês codificado com ASCII, podem armazenar páginas do manual (os diretórios **man<section>**) diretamente em **/usr/share/man**. (esse é na verdade uma circunstância tradicional e um arranjo.)

Países para os quais existe um bom aceite para os personagens do conjunto de códigos padrão pode omitir o campo **<character-set>**, mas é altamente recomendável que seja incluída, especialmente para países com vários padrões concorrentes.

---

12. Por exemplo, se /usr/local/man não tem páginas de manual na seção 4 (dispositivos), então /usr/local/man/man4 pode ser

omitido.
13. A principal exceção a esta regra é o Reino Unido, o que é “GB” na ISO 3166, mas “UK” para a maioria  endereços de e-mail.





| **Linguagem** | **Território** | **Tipo de Caractér** | **Diretório** |
|---|---|---|---|
| Inglês | — | ASCII | /usr/share/man/en |
| Inglês | Reino Unido | ISO 8859-15 | /usr/share/man/en_GB |
| Inglês | Estados Unidos | ASCII | /usr/share/man/en_US |
| Frances | Canada | ISO 8859-1 | /usr/share/man/fr_CA |
| Frances | França | ISO 8859-1 | /usr/share/man/fr_FR |
| Alemão | Germânia | ISO 646 | /usr/share/man/de_DE.646 |
| Alemão | Germânia | ISO 6937 | /usr/share/man/de_DE.6937 |
| Alemão | Germânia | ISO 8859-1 | /usr/share/man/de_DE.88591 |
| Alemão | Suíça | ISO 646 | /usr/share/man/de_CH.646 |
| Japonês | Japão | JIS | /usr/share/man/ja_JP.jis |
| Japonês | Japão | SJIS | /usr/share/man/ja_JP.sjis |
| Japonês | Japão | UJIS (ou EUC-J) | /usr/share/man/ja_JP.ujis |

Do mesmo modo, previsões devem ser feitas para as páginas de instruções que são dependentes da arquitetura, como documentação dos dispositivos drivers ou baixo nível de comandos de administração do sistema. Estes devem ser colocados sob um diretório <arch> no diretório apropriado **man/<section>**o, por exemplo, uma página de manual para o comando ctrlaltdel i386 (8)  comando pode ser colocado em **/usr/share/man/<locale>/man8/i386/ctrlaltdel.8**.

Páginas de manual para comandos e dados em **/usr/local** estão armazenadas em **/usr/local/man**. Páginas de manual para  X11R6 são armazenados em **/usr/X11R6/man**. Daqui resulta que todas as hierarquias de página de manual no sistema devem ter a mesma estrutura de **/usr/share/man**.

As seções de paginas **cat (cat<section>)** contendo entradas de página do manual formatadas,  também são encontrados dentro de subdiretórios do <mandir>/<locale>, mas não são necessários nem podem ser distribuídos no lugar de origem nroff  dos fontes das  páginas de manual.

As seções numeradas de "1" a "8" são tradicionalmente definidas. Em geral, o nome do arquivo de páginas do manual está localizado dentro de uma seção especial no final como **.<section>.**

Além disso, páginas específicas do manual de alguns grandes conjuntos de aplicativos têm um sufixo adicional anexada ao nome do arquivo de página do manual. Por exemplo, o manual de manipulação do sistema de e-mail MH deve ter mh anexado a todas as páginas do manual MH. Todas as páginas do manual do sistema X Window  deve ter um x anexado ao arquivo.

A prática de colocação de páginas do manual com línguas diferentes em subdiretórios adequados de /usr/share/man também aplica-se a outras hierarquias de página do manual, como **/usr/local/man** e **/usr/X11R6/man**. (Esta parte da  norma aplica-se também mais tarde na seção sobre a estrutura opcional **/var/cache/man**).

## 4.11.6 - /usr/share/misc – Dados de diversas arquiteturas independentes.

Este diretório contém diversos arquivos de arquitetura independente que não exigem um subdiretório separado  sob **/usr/share.**

## 4.11.6.1 – Opções específicas.

Os seguintes arquivos ou links simbólicos para os arquivos, deve estar no diretório /usr/share/misc, se o subsistema correspondente estiver instalado:

| **Arquivo** | **Descrição** |
|---|---|
| `ascii` | Tabela de definição de caracteres ASCII (opcional) |

| **Arquivo** | **Descrição** |
|---|---|
| `magic` | Lista padrão de números mágicos para o comandos de arquivo (opcional) |
| `termcap` | Base da dados para capacitação de terminais (opcional) |
| `termcap.db` | Base de dados para capacitação de terminais (opcional) |

Outros arquivos (aplicação específica) podem aparecer aqui, mas um distribuidor pode colocá-los em **/usr/lib** na sua discrição [14].

## 4.11.7 - /usr/share/sgml – Dados SGML (opcional).

### 4.11.7.1 – Propósito.

**/usr/share/sgml** contém arquivos independente da arquitetura usada pelas aplicações SGML, como catálogos ordinário (não os centralizados, veja **/etc/sgml**), DTDs, entidades, ou folhas de estilo.

### 4.11.7.2 – Opções específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar no diretório /usr/share/sgml, se o correspondente subsistema estiver instalado:

| **Diretório** | **Descrição** |
|---|---|
| `docbook` | DTD docbook (opcional) |
| `tei` | DTD tei (opcional) |
| `html` | DTD html (opcional) |
| `mathml` | DTD mathml (opcional) |

Outros arquivos que não são específicos para um determinado DTD pode residir em seu próprio subdiretório.

## 4.11.8 - /usr/share/xml – Dados XML (opcional).

### 4.11.8.1 – Propósito.

**/usr/share/XML** contém arquivos independente da arquitetura usada por aplicativos XML, tais como catálogos comuns (não os centralizada, veja **/etc/sgml**), DTDs, entidades, ou folhas de estilo.

### 4.11.8.2 – Opções específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar no diretório **/usr/share/xml**, se o correspondente subsistema estuver instalado:

| **Diretório** | **Descrição** |
|---|---|
| `docbook` | DTD XML docbook (opcional) |
| `xhtml` | DTD XHTML (opcional) |
| `mathml` | DTD MathML (opcional) |

14. Alguns arquivos incluem: airport, birthtoken, eqnchar, getopt, gprof.callg, gprof.flat, inter.phone, ipfw.samp.filters, ipfw.samp.scripts, keycap.pcvt, mail.help, mail.tildehelp, man.template, map3270, mdoc.template, Mais.Ajuda, na.phone, nslookup.help, operator, scsi_modes, sendmail.hf, style, units.lib, vgrindefs, vgrindefs.db, ZIPcodes





## 4.12 – /usr/src– Código fonte (opcional).

### 4.12.1 – Propósito.

Os códigos fonte podem ser colocados no lugar, neste subdiretório, apenas para fins de referência [15].

15. Geralmente, os fontes não devem ser construídos dentro desta hierarquia.



# Capítulo 5. – A hierarquia /var

## 5.1 – Propósito.

**/var** contém arquivos de dados variáveis. Isso inclui arquivos e diretórios de **spool**, administração e registro de dados (**logs**), e Arquivos transitórios e temporários.

Algumas partes do **/var** não são partilháveis entre sistemas diferentes. Por exemplo, **/var/log, /var/lock,** e **/var/run**. Outras partes podem ser compartilhadas, notadamente **/var/mail, /var/cache/man, /var/cache/fonts**, e **/var/spool/news**.

**/var** é especificada aqui, a fim de tornar possível montar **/usr** somente para leitura. Tudo o que uma vez entrou em **/usr** e que é escrito, durante uma operação do sistema (por oposição à instalação e manutenção de software) devem estar em **/var**.

Se **/var** não pode ser feita em uma partição separada, muitas vezes, é preferível mover **/var** para fora da partição raiz e dentro da partição **/usr**. (Às vezes isso é feito para reduzir o tamanho da partição raiz ou quando o espaço de execução é baixo na partição raiz.) Todavia, **/var** não deve ser ligada à **/usr** porque esta faz a separação do **/usr** e **/var** mais difícil e é susceptível de criar um conflito de nomes. Em vez disso crie um link, de **/var** para **/usr/var**.

Aplicações geralmente não devem adicionar diretórios de nível superior ao **/var**. Essas listas só devem ser adicionadas se elas têm algum sistema de implicação de tamanho, e em consulta com a lista de discussão FHS.

## 5.2 – Requisitos.

Os seguintes diretórios ou links simbólicos para diretórios, são necessários em **/var.**

| **Diretório** | **Descrição** |
|---|---|
| `Cachê` | Dados de aplicações de cache |
| `lib` | Informações de estados de variáveis |
| `local` | Dados varáveis para /usr/local |
| `lock` | Arquivos bloqueados |
| `log` | Log de arquivos e diretórios |
| `opt` | Dados varáveis para for /opt |
| `run` | Dados relevantes para processos em execução |
| `spool` | Dados de aplicações spool |
| `tmp` | Arquivos temporários preservados entre reinicializações |

Vários diretórios são “reservados” no sentido de que não devem ser utilizados de forma arbitrária por alguns novo pedido, uma vez que seria incompatível com o histórico e/ou práticas locais. Eles são:

`/var/backups`

```
/var/cron
/var/msgs
/var/preserve
```

## 5.3 – Opções específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar em **/var**, se o subsistema correspondente estiver instalado:





| **Diretório** | **Descrição** |
|---|---|
| account | Processos de contas de logs (opcional) |
| crash | Dumps de quedas do sistema (opcional) |
| games | Dados variáveis de jogos (opcional) |
| mail | Arquivos de usuários mailbox (opcional) |
| yp | Arquivos de base de dados de informações de serviços de redes (NIS) (opcional) |

## 5.4 – /var/account – Processos de contas de log (opcional).

## 5.4.1 – Propósito.

Este diretório contém os registros contabilísticos (logs) dos processo corrente ativo e os dados do processo e composição usado (como usados em alguns sistemas UNIX-like por **lastcomm** e **sa**).

## 5.5 – /var/cache – Dados de aplicações de cache.

## 5.5.1 – Propósito.

**/var/cache** destina-se a dados armazenados em cache de aplicativos. Esses dados são gerados localmente como resultado do tempo de processamento, entradas e saídas ou cálculos. A aplicação deve ser capaz de regenerar ou recuperar os dados. Ao contrário do **/var/spool,** os arquivos armazenados em cache pode ser removidos sem perda de dados. Os dados devem permanecer válidos entre chamadas da aplicação e reinicialização doo sistema.

Arquivos localizados em **/var/cache** podem ser expirados em um modo de aplicação específico, pelo administrador do sistema, ou ambos. O pedido deve ser sempre capaz de se recuperar da eliminação manual destes arquivos (geralmente por causa da falta de espaço em disco). Nenhum outro requisito é feito no formato de dados dos diretórios de cache.

### Fundamentação.

A existência de um diretório separado para dados armazenados em cache permite que os administradores do sistema possam definir discos diferentes e políticas de backup de outros diretórios em **/var.**

## 5.5.2 – Opções específicas.

**Diretório** **Descrição**

| | |
|---|---|
| fonts | Fontes gerados localmente (opcional) |
| man | Paginas de manuais formatadas localmente (opcional) |
| www | Proxy WWW proxy ou dados de cache (opcional) |
| <package> | Dados de cachê de pacotes específicos (opcional) |





## 5.5.3 – /var/cache/fonts – Fontes gerados localmente (opcional).

### 5.5.3.1 – Propósito.

O diretório **/var/cache/fonts** deve ser usado para armazenar os fontes dinamicamente criadas. Em particular, todos os fontes que são gerados automaticamente pelo **mktexpk** devem estar localizados em subdiretórios apropriadamente nominado do /var/cache/fonts [1].

### 5.5.3.2 – Opções específicas.

Outras fontes criados dinamicamente também podem ser colocados na árvore, sob subdiretórios apropriadamente nominado do **/var/cache/fonts.**

## 5.5.4 – /var/cache/man – Páginas de manuais formatadas localmente (opcional).

### 5.5.4.1 – Propósito.

Este diretório fornece uma localização padrão para os sites que oferecem uma leitura apenas da partição **/usr**, mas desejam permitir localmente o cache de páginas de manuais formatadas. Sites que montam **/usr** como escrita (por exemplo, instalações com um único usuário) podem escolher não usar **/var/cache/man** e podem escrever páginas formatadas de manuais nos diretórios **cat<section>** em **/usr/share/man** diretamente. Recomendamos que a maioria dos sites utilizem uma das seguintes opções em vez disso:

- Pré-formatar todas as páginas do manual juntamente com as versões não formatada.

- Não permitir o cache de páginas do manual formatadas, e requerer formatação do que será feito cada vez que uma página de manual levantada.

- Permitir cache local das páginas de manual formatados in **/var/cache/man**.

A estrutura de /var/cache/man precisa refletir tanto o fato de permitir a hierarquia de múltiplas páginas do manual e a possibilidade de suporte a vários idiomas.

Dada uma página de manual nãp formatada que normalmente aparece no **<path>/man/<locale>/man<section>**, o directório para colocar páginas de manual formatadas em **/var/cache/man/<catpath>/<locale>/cat<section>**, onde **<catpath>** é derivado do **<path>** para remover qualquer cabeça **usr** e/ou partes à direita dos componentes do caminho. (Note que o componente **<locale>** pode estar faltando.)[2].

Paginas de manual escritas para **/var/cache/mam** podem, eventualmente, ser transferida para diretórios pré-formatados adequados na hierarquia de origem do **man** ou expirado; do mesmo modo páginas de manual formatadas na fonte da hierarquia **man** podem ser expirados se eles não são acessados por um determinado período de tempo.

Se páginas de manual pré-formatadas vêm com um sistema em uma mídia somente leitura (um CD-ROM, por exemplo), devem ser instaladas na hierarquia **man** de origem (por exemplo **/usr/share/man/cat<section>**). **/var/cache/man** é reservado como um cachê de escrita para páginas de manuais formatadas.

---

1. Esta norma atualmente não incorpora a estrutura do diretório TeX (um documento que descreve o layout de arquivos e diretórios TeX), mas a leitura pode ser útil. Fica na **ftp://ctan.tug.org/tex/**
2. Por exemplo, /usr/share/man/man1/ls.1 é formatado em /var/cache/man/cat1/ls.1 e /usr/X11R6/man/<locale>/man3/XtClass.3x em /var/cache/man/X11R6/<locale>/cat3/XtClass.3x.





### Fundamentação.

A release 1.2 do padrão especificado **/var/catman** para esta hierarquia. O caminho (path) foi movido para dentro **/var/cache** para refletir melhor a natureza dinâmica das páginas man formatadas. O nome do diretório foi mudado para **man** para permitir o reforço da hierarquia para incluir outros formatos pós-processados como **"cat"**, além de outros como **PostScript, HTML** ou **DVI**.

## 5.6 – /var/crash – Dump de quedas do sistema (opcional).

## 5.6.1 – Propósito.

Este diretório contém **dumps** de memória do sistema. Até a data desta release da norma, **dumps** de memória do sistema não eram suportado no Linux, mas pode ser suportado por outros sistemas que possam cumprir com as FHS.

## 5.7 – /var/games – Dados variáveis de jogos (opcional).

## 5.7.1 – Propósito.

Todos os dados variáveis relativas aos jogos em **/usr** deve ser colocado aqui. **/var/games** deve conter os dados variável anteriormente encontrados em **/usr**; dados estáticos, como o texto de ajuda, descrições de nível, e assim por diante, devem permanecer em outro lugar, como **/usr/games/share**.

### Fundamentação.

Tem sido dada uma hierarquia própria para **/var/games** em vez de deixar fundir-se com o velho **/var/lib** como na versão 1.2. A separação permite o controle local das estratégias de backup, permissões e uso de disco, bem como permitir que host inter-partilhem reduzindo a desordem em **/var/lib**. Além disso, **/var/games** é o caminho tradicionalmente utilizados pelos BSD.

## 5.8 – /var/lib – Informações de estados variáveis.

### 5.8.1 – Propósito.

Essa hierarquia contém informações relativas ao estado de uma aplicação ou sistema. Estado informações são dados que modificam programas enquanto eles estão sendo executados, e que pertence a um host específico. Os usuários não devem precisar modificar arquivos em **/var/lib** para configurar um pacote em operação.





Informações de estado são geralmente usadas para preservar a condição de uma aplicação (ou um grupo de aplicações inter-relacionadas) entre invocações e entre diferentes instâncias do mesmo aplicativo. Informações de estado devem geralmente permanecem válidas após uma reinicialização, não deve ser a saída de **log**, e não devem ser dados no **spool**.

Uma aplicação (ou um grupo de aplicações inter-relacionadas) devem usar um subdiretório de **/var/lib** para colocar seus dados. É necessário um subdiretório, **/var/lib/misc**, que é destinado á arquivos de estado que não necessitem de um subdiretório; outros subdiretórios só devem estar presentes se o pedido em questão é incluído na distribuição [3]. **/var/lib/<name>** é o local que deve ser usado por toda a distribuição de pacotes de apoio. Diferentes distribuições podem utilizar nomes diferentes, é lógico.

### 5.8.2 – Requisitos.

Os seguintes diretórios ou links simbólicos para diretórios, são necessários em **/var/lib:**

| **Diretório** | **Descrição** |
|---|---|
| Misc | Dados de estados diversos |

### 5.8.3 – Opções específicas.

Os seguintes diretórios ou links simbólicos para diretórios, deve estar em /var/lib, se o correspondente subsistema estiver instalado:

| **Diretório** | **Descrição** |
|---|---|

<editor> Editor de arquivos de backups e estados (opcional)
<pkgtool> Arquivos de pacote de suporte (opcional)
<package> Dados de estado para pacotes e subsistemas (opcional)
**hwclock** Estado de diretório para hwclock (opcional)
**xdm** Dados variáveis do gerenciados de display (opcional)

## 5.8.4 – /var/lib/<editor> – Editor de arquivos de backup´s e estados (opcional).

## 5.8.4.1 – Propósito.

Estes diretórios contém arquivos salvos gerados por qualquer encerramento inesperado de um editor (por exemplo, **Elvis, jove, nvi**).

Outros editores podem não exigir um diretório para recuperação de crash (perda) de arquivos, mas pode exigir um local bem definido para armazenar outras informações, enquanto o editor está sendo executado. Estas informações devem ser armazenadas em um subdiretório **/var/lib** (por exemplo, GNU Emacs iria colocar os arquivos de lock em **/var/lib/emacs/lock**).

Futuros editores podem solicitar informações de estados adicionais para perda e recuperação de arquivos bloqueados – Estas informações também deve ser colocado dentro do **/var/lib/.**

### Fundamentação.

As versões anteriores do Linux, bem como todos os fornecedores comerciais, utilizam **/var/preserve** ou seus clones. No entanto, cada editor usa seu próprio formato para estes arquivos de perda-recuperação, apenas um diretório separado é necessário para cada editor.

---

3. Uma diferença importante entre esta versão deste padrão e as anteriores é que as aplicações são obrigadas a utilizar um subdiretório de /var/lib.





Editores específicos de arquivos de bloqueio são geralmente muito diferentes dos dispositivos ou arquivos de bloqueios de recursos que são armazenados em **/var/lock** e, portanto, são armazenados em **/var/lib**.

## 5.8.5 – /var/lib/hwclock – Diretório de estado para hwclock (opcional).

## 5.8.5.1 – Propósito.

Este diretório contém o arquivo **`/var/lib/hwclock/adjtime`**.

### Fundamentação.

No FHS 2.1, este arquivo era o / etc/adjtime, mas como atualização do hwclock, que era obviamente incorreta.

## 5.8.6 – /var/lib/misc – Dados variáveis diversos.

## 5.8.6.1 – Propósito.

Este diretório contém dados variáveis não colocados em um subdiretório em /var/lib. Uma tentativa deve ser feita para usar nomes relativamente único nesta pasta para evitar conflitos de nomes de espaços[4].

# 5.9 – /var/lock – Arquivos bloqueados.

## 5.9.1 – Propósito.

Arquivos bloqueados devem ser guardados dentro da estrutura de diretório **/var/lock.**

Arquivos de bloqueio de dispositivos e outros recursos compartilhados por vários aplicativos, como os arquivos de bloqueio serial dispositivo que eram originalmente encontrados em qualquer um dos dois diretório **/usr/spool/locks** ou **/usr/spool/uucp**, agora devem ser armazenados em **/var/lock**. A convenção de nomeação, que deve ser usado é "LCK .." seguido do nome base do dispositivo. Por exemplo, para bloquear **/dev/ttyS0,** o arquivo "LCK .. ttyS0" deve estar criado[5].

O formato usado para o conteúdo dos arquivos de bloqueio deve ser o formato de arquivo de bloqueio HDB UUCP. O formato HDB é para armazenar o identificador do processo (PID) como um número decimal de dez byte ASCII, com uma nova linha. Por exemplo, se o processo 1230 contém um arquivo de bloqueio, ele deverá conter os onze caracteres: espaço, espaço, espaço, espaço, espaço, espaço, espaço, espaço, espaço, um, dois, três, zero, e nova linha(CR).

---

4. Esta hierarquia deve conter os arquivos armazenados em /var/db nas versões atuais do BSD. Estes incluem **locate.database** e **mountdtab** e o banco de dados símbolo de kernel (s).
5. Então, qualquer coisa que desejar usar **/dev/ttyS0** pode ler o arquivo de bloqueio e agir em conformidade (em todos os bloqueios **/var/lock** deve ser legível para todo mundo).





# 5.10 – /var/log – Diretoriose arquivos de contabilização (logs).

## 5.10.1 – Propósito.

Este diretório contém diversos arquivos de contabilização (log). A maioria dos arquivos de logs devem ser escrito para este diretório ou um subdiretório adequado.

## 5.10.2 – Opções específicas.

Os seguintes arquivos ou links simbólicos para os arquivos, deve estar em **/var/log**, se o subsistema correspondente estiver instalado:

**Arquivo** **Descrição**

| | |
|---|---|
| lastlog | Registro do último login de cada usuário. |
| messages | Mensagens do sistema para **syslogd** |
| wtmp | Registro de todos os logins e logouts |

## 5.11 – /var/mail – Arquivos de usuários de caixa de e-mails (mailbox).

### 5.11.1 – Propósito.

As saídas de correios(e-mails) devem ser acessíveis através do **/var/mail** e os arquivos de **spool** de email deve assumir a forma **<username>.** [6]

Arquivos da caixa de correios de usuários (e-mails) neste local devem ser armazenados no formato padrão de caixa de correios (mailbox) UNIX.

### Fundamentação.

O local lógico deste diretório foi alterado a partir de **/var/spool/mail** a fim de trazer FHS em linha com quase toda implementação de Unix. Essa mudança é importante para a inter-operabilidade a partir um único **/var/mail** é frequentemente compartilhada entre vários hosts e várias implementações UNIX (apesar de problemas de bloqueio no NFS).

É importante notar que não há nenhuma obrigação de mover fisicamente o **spool** de **email** para este local. No entanto, programas e arquivos de cabeçalho deve ser alterado para usar **/var/mail**.

## 5.12 – /var/opt – Dados variáveis para /opt.

### 5.12.1 – Propósito.

Dados variáveis dos pacotes em **/opt** deve ser instalado em **/var/opt/<subdir>**, onde **<subdir>** é o nome da subárvore em **/opt** onde os dados estáticos de uma adição de um pacote de software, salvo se substituído por outro arquivo em **/etc** Nenhuma estrutura é imposta sobre o arranjo interno de **/var/opt/<subdir>**.

6. Note-se que /var/mail pode ser um link simbólico para outro diretório.





### Fundamentação.

Consulte a documentação e fundamentação para /opt.

## 5.13 – /var/run – Dados variáveis em tempo de execução (run-time).

### 5.13.1 – Propósito.

Este diretório contém as informações dos dados do sistema que descreve o sistema desde que foi iniciado. Arquivos que estão dentro desse diretório devem ser inicializados (removido ou limpos conforme o caso), no início do processo de inicialização. Programas podem ter um subdiretório de **/var/run**, isto é incentivado para programas que usam mais de arquivo de **run_time**. [7]
Arquivos de identificador de processo (PID), que foram originalmente colocados em **/etc**, devem ser colocados agora em **/var/run**. A convenção de nomes para arquivos PID é **<program-name>.pid.** Por exemplo, o arquivo PID **crond** é nomeado para **/var/run/crond.pid**.

## 5.13.2 – Requisitos.

O formato interno dos arquivos PID permanecem inalterados. O arquivo deve consistir do identificador do processo em ASCII decimal codificado, seguido por um caractere de nova linha(CR). Por exemplo, se **crond** foi processo com o número 25, /var/run/crond.pid deveria conter três caractéres: dois, cinco, e de nova linha(CR).

Programas que leem arquivos PID devem ser um pouco flexíveis no que eles aceitam, ou seja, eles devem ignorar espaços extra, zeros à esquerda, a ausência de uma nova linha de fuga(CR), ou linhas adicionais no arquivo PID. Programas que criam arquivos PID devem utilizar a especificação simples localizada no parágrafo acima.

O arquivo **utmp**, que armazena informações sobre quem está usando o sistema, está localizado neste Diretório.

Programas do sistema que mantém portas (sockets) de domínio transitório do UNIX, devem colocá-las neste diretório

## 5.14 – /var/spool – Dados de aplicações de saída(spool).

## 5.14.1 – Propósito.

/var/spool contém dados que estão à espera de algum tipo de processamento posterior. Dados em /var/spool representam trabalho a ser ser feito no futuro (por um programa, usuário ou administrador), muitas vezes os dados são apagados depois de terem sido processados. [8]

## 5.14.2 – Opções específicas.

Os seguintes diretórios ou links simbólicos para diretórios, devem estar em /var/spool, se o correspondente subsistema estiver instalado:

---

7.-**/var/run** devem estar protegidos para usuários sem privilégios **(root ou usuários daemons são execução)**, é um importante problema de segurança se qualquer usuário puder escrever neste diretório.

8. arquivos bloqueados UUCP devem ser colocados em **/var/lock**. Veja a secção acima sobre **/var/lock.**





| **Diretório** | **Descrição** |
|---|---|
| `lpd` | Diretório de Spool de impressão (opcional) |
| `mqueue` | Fila de envio de e-mail (opcional) |
| `news` | Novo diretório de Spool (opcional) |

| | |
|---|---|
| rwho | Arquivos Rwhod (opcional) |
| uucp | Diretório de spool para UUCP (opcional) |

## 5.14.3– /var/spool/lpd – Filas de impressão daemon para impressoras em linha (opcional).

### 5.14.3.1 – Propósito.

O arquivo de bloqueio para **lpd**, lpd.lock,deve ser colocado em **/var/spool/lpd**. Sugere-se que o arquivo de bloqueio para cada impressora deve ser colocado no diretório de spool para que a impressora específica, bloqueie e chame o arquivo para ser impresso.

### 5.14.3.2 – Opções específicas.

| Diretório | Descrição |
|---|---|
| printer | Arquivos de Spools para uma impressora específica (opcional) |

## 5.14.4– /var/spool/rwho – Arquivos rwhod (opcional).

### 5.14.4.1 – Propósito.

Este diretório contém as informações rwhod para outros sistemas na rede local.

### Fundamentação.

Alguns versões BSD usam **/var/rwho** para estes dados, devido à sua localização histórica in **/var/spool** em outros sistemas e seu encaixe aproximado com a definição de ddos **"spooled"**, este local foi considerado mais apropriado.

## 5.15 – /var/tmp – Arquivos temporários preservados entre reinicializações do sistema.

### 5.15.1 – Propósito.

O diretório **/var/ tmp** está disponível para programas que exigem arquivos temporários ou pastas que são preservados entre reinicializações do sistema. Portanto, os dados armazenados em /var/tmp são mais persistentes do que os dados em /tmp.

Arquivos e diretórios localizados em **/var/tmp** não devem ser excluídos quando o sistema for iniciado. Embora dados armazenados em **/var/tmp** são normalmente excluídos de uma maneira específica, é recomendado que as remoções ou deleções ocorram em um intervalo menor de freqüência do que em **/tmp**.

## 5.16 – /var/yp – Arquivos de base de dados para Serviços de informações de rede (Network Information  Service - NIS) (opcional).

## 5.16.1 – Propósito.

Dados variáveis para o serviço de informações de redes - Network Information Service (NIS), anteriormente conhecido como Sun Yellow Pages (YP), deve  ser colocadas neste diretório.

### Fundamentação.

**/var/yp** é o diretório padrão para dados de NIS (YP) e é utilizado quase exclusivamente na documentação NIS e sistemas.[9]

9.-NIS não deve ser confundida com a Sun NIS, que usa um diretório diferente, **/var/nis.**



# Capítulo 6. – Anexo específico do Sistema Operacional

Esta seção é para os requisitos adicionais e recomendações que se aplicam apenas a um sistema operacional específico. O material desta seção nunca deve entrar em conflito com a norma básica padrão.

## 6.1 – Linux.

Esse é um anexo para o sistema operacional Linux.

## 6.1.1 – / : Diretório root.

Em sistemas Linux, se o kernel está localizado em **/**, recomendamos o uso dos nomes **vmlinux** ou **vmlinuz,** que têm sido utilizados nos últimos pacotes fonte do kernel Linux.

## 6.1.2 – /bin : Comandos binários essenciais de usuários (para ser usado por todos os usuários) .

Existem sistemas Linux que obrigam colocar estes arquivos adicionais em **/bin**:

- **setserial**

## 6.1.3 – /dev : Arquivos especiais e dispositivos .

Os seguintes dispositivos devem estar dentro do **/dev**.

**/dev/null**

Todos os dados escritos para este dispositivo são descartados. Uma leitura a partir deste dispositivo retornará uma condição EOF.

**/dev/zero**

Este dispositivo é uma fonte de dados zerado. Todos os dados escritos para este dispositivo são descartados. Uma leitura a partir deste dispositivo retornará como resultado, muitos bytes contendo o valor zero como foi solicitado.

**/dev/tty**

Este dispositivo é um sinônimo para controlar o terminal de um processo. Uma vez que este dispositivo é aberto, todas as leituras e escritas vão se comportar como se próprio terminal de controle tinha sido aberto.

### Fundamentação.

Versões anteriores do FHS tinha requisitos mais rigorosos para **/dev**. Outros dispositivos também podem existir em **/dev**. Os nomes dos dispositivos podem existir como links simbólicos para outro dispositivo ou nós localizados em **/dev** ou subdiretórios do **/dev**. Não há nenhuma exigência relativa a valores de números maior/menor.

## 6.1.4 – /etc : Configurações específicas do sistema local – Host específico.

Existem sistemas Linux que os obrigam a colocar estes arquivos adicionais em /etc

- `lilo.conf`

## 6.1.5. /lib64 e /lib32 : Bibliotecas para 64/32-bits (arquitetura dependente)

As arquitetura de 64-bits PPC64, s390x, sparc64 e AMD64 devem colocar bibliotecas 64-bit em **/lib64**, e bibliotecas de 32-bit (ou 31-bit em s390) em **/lib**.

A arquitetura de 64-bitIA64 deve colocar bibliotecas 64-bit em **/lib**.

### Fundamentação.

Este é um refinamento das regras gerais para **/lib<qual>** e **/usr/lib<qual>**. As arquiteturas PPC64, s390x, sparc64 e que suportam AMD64, suportam tanto programas, de 32-bit (para s390 mais precisos 31-bit) como programas de 64-bits.

Usando lib para binários de 32 bits, permite que sistemas binários existentes a partir do 32-bit possam funcionar sem qualquer alteração: Espera-se que binários sejam numerosos. IA-64 usa um esquema diferente, refletindo a depreciação de binários de 32-bits (e, portanto, bibliotecas) na arquitetura.

## 6.1.6. /proc : Kernel e informações de processos do sistema de arquivo virtual.

O sistema de arquivos **proc** é de fato o método padrão do Linux para o tratamento de processos e informações do sistema, em vez de /dev/kmem e outros métodos similares. Encorajamos fortemente o uso disso para o armazenamento e recuperação de informações de processos, bem como outro kernel e informações de memória memória.

## 6.1.7. /sbin : Sistema binário essencial.

O sistemas Linux coloca estes arquivos adicionais em **/sbin**.

- Comandos estendidos do sistema de arquivos (opcional)
  - **badblocks**
  - **dumpe2fs**
  - **e2fsck**
  - **mke2fs**
  - **mklost+found**
  - **tune2fs**
- Mapa de instalação da inicialização (do boot-loader) (opcional)
  - **lilo**

Arquivos opcionais para **/sbin:**

- Binários estáticos:
  - **ldconfig**
  - **sln**
  - **ssync**

Estáticos **ln** (s**ln**) e **sync** (s**sync**) são úteis quando as coisas dão errado. A principal utilização do s**ln** (reparar links simbólicos incorretos em **/lib** depois de uma atualização mal orquestrada) não é mais uma preocupação maior, agora que o programa **ldconfig** (usualmente localizado em **/usr/sbin**) existe e pode agir como um fio condutor na melhoria de bibliotecas dinâmicas. Estático **Sync** é útil em algumas situações de emergência. Note que os mesmos não precisam ser estaticamente ligados as versões da **ln** e **sync** padrão, mas pode ser.

O binário **ldconfig** é opcional em **/sbin** uma vez que um site pode escolher executar **ldconfig** no momento da inicialização (boot time), ao invés de somente para atualizar as bibliotecas compartilhadas. (Ainda não está claro se é ou não vantajoso executar **ldconfig** em cada boot). Mesmo assim, algumas pessoas gostam do **ldconfig** nas seguintes (muito comuns) situações:

1. Eu apenas removi /lib/<file>.

2. Eu não posso descobrir o nome da biblioteca porque ela está dinamicamente relacionada, eu estou usando um shell que não pode ter "**ls built-in**", e eu não sei como usar o comando "**echo** *" como um substituto.

3. Eu tenho um s**ln** estático, mas eu não sei como chamar o link.

Diversos

- **ctrlaltdel**
- **kbdrate**

Para lidar com o fato de que alguns teclados venham com uma taxa de repetição elevada, para isso ser inutilizável, kbdrate pode ser instalado em **/sbin** em alguns sistemas.

Desde a ação padrão no kernel para a combinação **Ctrl+Alt+Del** ser a chave para reiniciar o equipamento instantaneamente, é geralmente aconselhável desativar o comportamento antes de montar o sistema de arquivos raiz em modo **read-write**. Alguns suítes **init** são capazes de desabilitar o **Ctrl+Alt+Del**, mas outros podem exigir que o programa **ctrlaltdel**, que pode ser instalado em **/sbin** estejam nesses sistemas.

## 6.1.8. /usr/include : Arquivos de cabeçalho incluídos por programas C.

Esses links simbólicos são exigidos se um compilador C ou C++ estiver instalado e apenas para sistemas não baseados em **glibc.**

```
/usr/include/asm -> /usr/src/linux/include/asm-<arch>
```

```
/usr/include/linux -> /usr/src/linux/include/linux
```

## 6.1.9. /usr/src : Código fonte.

Para sistemas baseados em **glibc**, não há orientações específicas para este diretório. Para sistemas baseados em Linux libc com revisões antes da **glibc**, as seguintes diretrizes e fundamentos se aplicam:

O único código fonte que deve ser colocado em um local específico é o código fonte do kernel Linux. Ele está localizado em **/usr/src/linux.**





Se um compilador C ou C++ está instalado, mas código-fonte completo do kernel Linux não está instalado, então os arquivos incluídos a partir do código-fonte do kernel devem estar localizados nesses diretórios:

```
/usr/src/linux/include/asm-<arch>

/usr/src/linux/include/linux
```

**<arch>** é o nome da arquitetura do sistema.

**Nota:**

/usr/src/linux deve ser um link simbólico para uma árvore de código fonte do kernel..

### Fundamentação.

É importante que os arquivos de inclusão do kernel estejam localizados em **/usr/src/linux** e não no **/usr/include** assim não haverá problemas quando o administrador do sistema atualizar sua versão do kernel pela primeira vez.

## 6.1.10. /var/spool/cron : cron e para Jobs.

Este diretório contém os dados variáveis para o cron e para os programas.

# Capítulo 7. – Apêndice.

## 7.1. - A lista de discussão do FHS.

A lista de discussão FHS está localizado na **<freestandards-fhs-discuss@lists.sourceforge.net>**. Você poderá se inscrever na lista de discussão nesta página: **http://sourceforge.net/projects/freestandards/**.

Agradecemos a rede de operações da Universidade da Califórnia em San Diego, que nos permitiu usar seus excelentes servidores de listas de correios (mailing list).

Conforme observado na introdução, por favor não envie e-mail para a lista de discussão sem antes contatar o editor FHS ou um colaborador listado.

## 7.2. - A retaguarda do FHS.

O processo de desenvolvimento de um padrão de sistema de hierarquia de arquivos começou em agosto de 1993, com um esforço para reestruturar a estrutura de arquivos e diretórios do Linux. O FSSTND, um padrão de sistema de hierarquia de arquivos específico para o sistema operacional Linux, foi lançado em 14 de fevereiro de 1994. Foram lançadas revisões em 9 de outubro de 1994 e 28 de março de 1995. No início de 1995, o objetivo de desenvolver uma versão mais abrangente do FSSTND para tratar não só do Linux, mas também de outros sistemas UNIX foi aprovada com a ajuda de membros da comunidade de desenvolvimento do BSD. Como resultado, um esforço foi feito para se concentrar em questões que foram gerais para sistemas UNIX-like. Em reconhecimento deste alargamento do âmbito de aplicação, o nome do padrão foi alterado para Filesystem Hierarchy Standard ou FHS como abreviatura.

Voluntários que tem contribuíido amplamente para esta norma são listadas no final deste documento. Esta norma representa uma visão de consenso dessas pessoas e outros colaboradores.

## 7.3. Orientações gerais.

Aqui estão algumas das orientações que têm sido utilizadas no desenvolvimento desta norma:

Resolver problemas técnicos, limitando dificuldades transitórias.

Fazer a especificação razoavelmente estável.

Ganhar a aprovação dos distribuidores, desenvolvedores e outros tomadores de decisão de grupos de desenvolvimento relevantes e incentivar a sua participação.

Fornecer um padrão que seja atraente para os implementadores de diferentes sistemas UNIX-like.

# 7.4. Escopo.

Este documento especifica padrão de sistema de hierarquia de arquivos para sistemas de arquivos FHS, especificando a localização dos arquivos e diretórios, e o conteúdo de alguns arquivos de sistema.

Esta norma foi concebida para ser utilizada pelos integradores de sistemas, desenvolvedores de pacote, e administradores de sistemas na construção e manutenção de sistemas de arquivos FHS compatível. É essencialmente destinado a ser um referência e não um tutorial sobre como gerenciar um sistema de arquivos conforme a hierarquia.





O FHS cresceu a partir de trabalhos anteriores sobre FSSTND, um padrão de organização de sistema de arquivos para o sistema operacional Linux. Baseia-se na FSSTND para resolver problemas de interoperabilidade não apenas na comunidade Linux, mas em uma ampla área, incluindo sistemas operacionais baseados em 4.4BSD. Ele incorpora as lições aprendidas no mundo BSD e outras sobre o suporte multi-arquitetura e as demandas de redes heterogêneas.

Embora esta norma é mais abrangente do que as tentativas anteriores de padronização do sistema de hierarquia de arquivos, atualizações periódicas podem ser necessárias como requisitos de mudança em relação à tecnologia emergente. É também possível que melhores soluções para os problemas abordados aqui sejam descobertos o que fará com que as nossas soluções não serão mais as melhores soluções possíveis. Projetos complementares podem ser liberados, além de atualizações periódicas para este documento. No entanto, a compatibilidade de um objetivo específico está por trás de uma versão deste documento para a próxima.

Comentários relacionados com esta norma são bem-vindos. Quaisquer comentários ou sugestões para mudanças podem ser dirigidas para o editor FHS (Daniel Quinlan <quinlan@pathname.com>) ou a lista de discussão FHS Comentários tipográficos ou gramaticais devem ser dirigidos ao editor FHS.

Antes de enviar e-mail para a lista de discussão é solicitado que você primeiro contato com o editor FHS, a fim de evitar excessiva re-discussão de temas antigos.

Perguntas sobre como interpretar os artigos deste documento pode ocasionalmente surgir. Se você tiver necessidade de um esclarecimentos, entre em contato com o editor de FHS. Uma vez que este padrão representa um consenso de muitos participantes, é importante ter certeza de que qualquer interpretação também representa a opinião coletiva deles. Por esta razão, pode não ser possível dar uma resposta imediata a menos que a dúvida tenha sido objeto de discussão anterior.

# 7.5. Agradecimentos.

Os desenvolvedores da FHS querem agradecer aos desenvolvedores, administradores de sistema e os usuários cuja entrada foi essencial para esta norma. Queremos agradecer a cada um dos colaboradores que ajudaram a escrever, compilar e compor esta norma.

O Grupo FHS também agradece a todos os desenvolvedores Linux que apoiaram a FSSTND, o predecessor desta norma. Se eles não tivessem demonstrado que a FSSTND foi benéfico, o FHS nunca poderia ter evoluído.

## 7.6. Colaboradores.

Brandon S. Allbery <bsa@kf8nh.wariat.org>
Keith Bostic <bostic@cs.berkeley.edu>
Drew Eckhardt <drew@colorado.edu>
Rik Faith <faith@cs.unc.edu>
Stephen Harris <sweh@spuddy.mew.co.uk>
Ian Jackson <ijackson@cus.cam.ac.uk>
Andreas Jaeger <aj@suse.de>
John A. Martin <jmartin@acm.org>
Ian McCloghrie <ian@ucsd.edu>
Chris Metcalf <metcalf@lcs.mit.edu>
Ian Murdock <imurdock@debian.org>
David C. Niemi <niemidc@clark.net>
Daniel Quinlan <quinlan@pathname.com>
Eric S. Raymond <esr@thyrsus.com>
Rusty Russell <rusty@rustcorp.com.au>





Mike Sangrey <mike@sojurn.lns.pa.us>
David H. Silber <dhs@glowworm.firefly.com>
Thomas Sippel-Dau <t.sippel-dau@ic.ac.uk>
Theodore Ts'o <tytso@athena.mit.edu>
Stephen Tweedie <sct@dcs.ed.ac.uk>
Fred N. van Kempen <waltje@infomagic.com>
Bernd Warken <bwarken@mayn.de>
Christopher Yeoh <cyeoh@samba.org>